(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) Publication number: **0 492 901 A2**

(12) **EUROPEAN PATENT APPLICATION**

(21) Application number: **91311524.2**

(51) Int. Cl.5: **G06F 15/72**

(22) Date of filing: **11.12.91**

(30) Priority: **27.12.90 JP 415177/90**

(43) Date of publication of application:
01.07.92 Bulletin 92/27

(84) Designated Contracting States:
CH DE ES FR GB IT LI NL SE

(71) Applicant: **International Business Machines Corporation**
Old Orchard Road
Armonk, N.Y. 10504(US)

(72) Inventor: **Hidaka, Kazuyoshi**
2-177 Fujidana-cho, Nishi-ku
Yokohama-shi, Kanagawa-ken(JP)

(74) Representative: **Blakemore, Frederick Norman**
IBM United Kingdom Limited Intellectual Property Department Hursley Park
Winchester Hampshire SO21 2JN(GB)

(54) **Apparatus for and method of handling graphics objects.**

(57) A graphics system comprises a graphics object table 10 that stores, for each graphics object, its identifier and the identifiers of one or more control points defining the graphics object; a control point table 11 that stores, ~~for each control point,~~ its identifier and positional data, ~~said positional data being permitted to be data on the~~ position of the control point relative to a graphics object; and means of updating the graphics object table 10 and the control point table 11 in response to the setting of one or more control points for a new graphics object by the user. These tables are referred to in response to a movement or reshaping of a primary graphics object, and the coordinate values of the affected control points are calculated by using a coordinate calculating unit 7.

Even when a number of graphics objects are interrelated, a graphics object under a geometical constraint exerted directly or indirectly by another graphics object is redrawn in response to the movement or reshaping of that object. Further, such redrawing can be done in response to a movement or reshaping other than translation, rotation, or scaling.

EP 0 492 901 A2

BEST AVAILABLE COPY

The present invention relates to apparatus for and a method of handling graphics objects, which generate a new graphics object geometrically constrained by a previously generated graphics object, and, when a graphics object is reshaped or moved, permit another graphics object generated under a geometrical constraint exerted directly or indirectly by the reshaped or moved graphics object to be redrawn in response to said reshaping or movement. Graphics objects include objects with dimensions from zero (points) to three (surfaces and so forth).

In many commercially available computer-based graphics object handling systems, points are set on a display screen in order to generate and display a graphics object. For instance to draw a segment, one must set its two ends, and to draw a circle, one must set the center of the circle and a point on its circumference. Such points defining a graphics object are called control points.

When a user intends to draw a new graphics object on a display screen, he often sets control points for the new graphics object while recognizing a geometry exerted by a graphics object that has previously been drawn. Hereafter, when, a graphics object is generated in a geometric relation to a previously generated graphics object, the latter is called a primary graphics object relative to the former, and the former is called a secondary graphics object relative to the latter, for simplicity.

During the reshaping or the movement of a primary graphics object, however, commercially available prior art systems for handling graphics objects cannot redraw a secondary graphics object automatically in response.

As one solution to the problem JA PUPA 1-246673 proposes a graphics object handling system in which, for each graphics object, identifiers of other graphics objects to be connected to the graphics object are stored in a memory, and, when one of the graphics objects is edited and changed, geometrical data on other graphics objects connected to it are modified in such a way that the connection between them is maintained. The system, however, involves the following problems:

(1) Since the system stores only the IDs of other graphics objects to be connected as connection information, it may register that a graphics object has been redrawn, but is unable to determine which control point for said graphics object should have its coordinate values changed. Connections among graphics objects are not always trivial. A single graphics object may be defined by a number of control points; a number of graphics objects may be connected to a single graphics object; or a number of graphics objects may be linked in a chain. In order to reflect a change in the position of a graphics object relative to other graphics objects, a technique for extracting affected control points is indispensable. The above system is deficient in this respect, and cannot propagate a change in the position of one of several intricately connected graphics objects to the others.

(2) The above system, which might be able to redraw a secondary object in response to the translation, rotation, magnification or reduction of a primary graphic object, cannot cope with other movements or reshaping.

In summary, the problems in the prior art stem from the fact that only the coordinate values of control points are given as their positional data. As a result, the system cannot calculate the new coordinate values of control points and therefore cannot redraw a secondary graphics object in response to the reshaping or the movement of a primary graphics object.

In accordance with the present invention, there is provided apparatus for generating a graphics object by setting one or more control points, the apparatus comprising:

(a) a graphics object table that stores, for each graphics object, its identifier and identifiers of one or more control points;

(b) a control point table that stores, for each control point, its identifier and positional data, the positional data being permitted to be data on the position of the control point relative to a graphics object; and

(c) means of updating the graphics object table and the control point table in response to the setting of one or more control points for a new graphics object by a user.

Thus the present invention advantageously provides apparatus for handling graphics objects that can describe the positions of control points for a graphics object in a novel form that is not restricted to the coordinate values of control points.

Preferably, the apparatus comprises means for displaying a graphics object that has been or is being generated.

The data on the relative position of said control point are specifically as follows: First, if said control point is set upon a control point for a graphics object, they consist of the identifier of the control point. If said control point is set upon an intersection point defined by two graphics objects, they are the identifiers of the two graphics objects. If the graphics objects defining the intersection point are a quadratic curve and a line, or if both are quadratic curves, the identification data for the intersection point are included in them. If said control point is set upon an internal division point of an existing segment, they are the identifier of the

BNSDOCID: <EP 0492901A2 I >

segment object and the ratio of the internal division. If the control point is set upon a point of the circumference, they are the identifier of the circular object and an angle.

Thus, the present invention advantageously facilitates operations for generating a graphics object that is geometrically constrained by another graphics object.

Viewing a second aspect of the present invention, there is provided apparatus for redrawing a graphics object generated under a geometrical corstraint exerted directly or indirectly by another graphics object in response to the reshaping or movement of the latter, the apparatus comprising:

(a) a graphics object table that stores, for each graphics object, its identifier and the identifiers of one or more control points;

(b) a control point table that stores, For each control point, its identifier and positional data, the positional data being permitted to be data on the position of the control point relative to a graphics object;

(c) means of referring to the graphics object table and the control point table, in response to the reshaping or movement of a graphics object in order to extract the identifiers of graphics objects that may have been generated under a geometrical constraint exerted directly or indirectly by the graphics object;

(d) means for calculating the coordinate values of control points which are for the extracted graphics objects and whose positional data are related to graphics objects, by utilizing said relative positional data; and

(e) means for displaying a graphics object defined by a control point at least one calculated coordinate value of which is different from the original value.

Thus the present invention provides a graphics object handling apparatus in which the relationships between one graphics object and another are described not merely by means of their identifiers as in the prior art, but in a novel and advantageous form.

Viewing a third aspect of the present invention, there is provided a method for generating a graphics object by setting one or more points to define the graphics object (hereafter called control points), comprising the steps of:

(a) making a graphics object table for storing, for each graphics object, an identifier thereof and identifiers of one or more control points;

(b) making a control point table that stores, for each control point, an identifier and positional data thereof, said positional data on the position being permitted to be data on the position of said control point relative to a graphics object; and

(c) updating said graphics object and said control point table in response to the setting of one or more control points for a new graphics object by a user.

The present invention permits a graphics object generated under a geometrical constraint exerted directly or indirectly by another graphic object to be redrawn in response to the movement or the reshaping of that other object while maintaining the constraint relationship between them, even when a number of graphics objects are intricately interrelated.

Viewing a fourth aspect of the present invention, there is provided in a graphics object handling system having a graphics object table that stores, for each graphics object, an identifier thereof and identifiers of one or more points defining the graphics object (hereafter called control points) and a control point table that stores, for each said control point, an identifier and positional data thereof, said positional data being permitted to be data on the position of the control point relative to a graphics object,

a method of redrawing a graphics object generated under a geometrical constraint exerted directly or indirectly by another graphics object in response to the reshaping or movement of the latter, comprising:

(a) in response to the reshaping or movement of a graphics object, referring to said graphics object table and said control point table to extract the identifiers of graphics objects that may have been generated under a geometrical constraint exerted directly or indirectly by the graphics object;

(b) calculating the coordinate values of control points which are for the extracted graphics objects and whose positional data are related to graphics objects, by utilizing said relative positional data; and

(c) displaying a graphics object defined by a control point at least one calculated coordinate value of which is different from the original value.

This advantageously permits a graphics object generated under a geometrical constraint exerted by a particular graphics object to be redrawn in response to a movement or reshaping of that other graphics object in a form other than translation, rotation, or scaling.

An embodiment of the present invention will now be described with reference to the accompanying drawings in which:

Figure 1 is an explanatory view of the definition of a graphics object using control points;

Figure 2 is an explanatory view of the definition of a graphics object using control points;

BNSDOCID: <EP 0492901A2 I >

Figure 3 is an explanatory view of prior art example 1;
Figure 4 is an explanatory view of a problem in prior art example 1;
Figure 5 is an explanatory view of prior art example 2;
Figure 6 is an explanatory view of the connection information used in prior art example 2;
Figure 7 is an explanatory view of how the graphics objects shown in Figure 5 are reshaped according to prior art example 2;
Figure 8 is an explanatory view of the graphics object reshaping permitted by prior art example 2;
Figure 9 is a view of the graphics object reshaping permitted by prior art example 2;
Figure 10 is an explanatory view of a problem in prior art example 2;
Figure 11 is a block diagram showing the components of an embodiment of the invention;
Figure 12 is a view showing an example of graphics objects handled by the embodiment;
Figure 13 is a view showing a display screen of the embodiment during operation;
Figure 14 is a view showing an example of how a secondary graphics object is made;
Figure 15 is an explanatory view of a control point effective range;
Figure 16 is a view showing an example of how a secondary graphics object is made;
Figure 17 is a view showing an example of how a secondary graphics object is made;
Figure 18 is an explanatory view of an intersection point effective range;
Figure 19 is an explanatory view showing an example of how a secondary graphics object is made;
Figure 20 is a view showing a point of intersection of a circle and a segment;
Figure 21 is an explanatory view of a problem that occurs when a control point is a point of intersection of a circle and a segment;
Figure 22 is an explanatory view of a fresh mode of an intersection point identifying unit;
Figure 23 is an explanatory view of a continuous mode of the intersection point identifying unit;
Figure 24 is an explanatory view of the principle of identification of an intersection point;
Figure 25 is an explanatory view of the principle of identification of an intersection point:
Figure 26 is an explanatory view of the operation of the intersection point identifying unit;
Figure 27 is an explanatory view of the operation of the intersection point identifying unit;
Figure 28 is an explanatory view of the principle of identification of an intersection point;
Figure 29 is an explanatory view of the principle of identification of an intersection point;
Figure 30 is a view showing an example of how a secondary graphics object is made;
Figure 31 is a view showing an example of how a secondary graphics object is made;
Figure 32 is a view showing an example of how a secondary graphics object is made;
Figure 33 is a view showing an example of how a secondary graphics object is made;
Figure 34 is a flow chart showing the operation of a coordinate calculating unit;
Figure 35 is a view showing the display screen before a primary graphics object is reshaped;
Figure 36 is a flow chart showing the redrawing process;
Figure 37 is a view showing the display screen after a primary graphics object has been reshaped;
Figure 38 is a view showing the display screen after a primary graphics object has been reshaped:
Figure 39 is a view showing the display screen after a primary graphics object has been reshaped; and
Figure 40 is a view showing the display screen after a primary graphics objects has been moved.

As aforesaid, in many commercially available computer-based graphics object handling systems, points are set on a display screen in order to generate and display a graphics object. For instance, referring to Figure 1, to draw a segment, one must set the centre of the circle and a point on its circumference. Such points defining a graphics object are called control points.

During the reshaping or the movement of a primary graphics object, as aforesaid, commercially available prior art systems for handling graphics objects cannot redraw a secondary graphics object automatically in response.

For instance, referring to Figure 3 to draw a segment L connecting vertices A and D, one may set control points for segment L upon points A and D. In this case, triangles ABC and DEF are primary graphics objects, whereas segment L is a secondary graphics object. The control points for segment L are points A and D. However, referring to Figure 4, when triangle ABC is reshaped (the position of point A is assumed to move) segment L nevertheless remains in the original position, so that it is not redrawn to connect vertices A and D after triangle ABC has been reshaped.

As aforementioned, JA PUPA 1-246673 proposes a graphics object handling system in which, for each graphics object, identifiers of other graphics objects to be connected to the graphics object are stored in a memory, and, when one of the graphics objects is edited and changed, geometrical data on other graphics objects connected to it are modified in such a way that the connection between them is maintained. For example, referring to Figures 5 and 6, when segments 33 and 34 connected respectively to rectangles 31

BNSDOCID: <EP 0492901A2 >

and 32 are displayed, ID data on connected graphics objects are stored in the memory as shown in Figure 6. Referring to Figure 7, if graphics object 32 is moved, geometrical data on graphics objects 33 and 34 connected to it are modified, and graphics objects 33 and 34 are redrawn so as to maintain the connection among graphics objects 32, 33 and 34.

Referring again to Figure 5, it is known that graphics objects 33 and 34 are changed in response to the movement of graphics object 32. However, this prior art does not disclose a method for determining which of control points e, f, g and h for segments 33 and 34 should have its coordinates correspondingly changed. If data on the coordinates of points e, f, g and h and geometric data on object 32 are examined, it may be found that points g and h are located on the object 32, which means that their coordinates need to be changed. With this method, however, if objects 32 and 31 are connected indirectly via other objects and hence if the coordinates of points e and f also need to be changed, points e and f are not recognised as needing to be changed.

Furthermore, referring to Figure 8, assume that segment L has been drawn by selecting the midpoint P3 of side P1 P2 of a triangle and P4 as control points. If the triangle is translated, the new coordinates of its control points are determined by simply translating point P3. Therefore, referring to Figure 9, even with the above system, it might be possible to redraw segment L. If, however, the triangle is reshaped by moving the point P1 alone, referring to Figure 10, it is impossible to redraw segment L. This is because data on the intended new location of point P3 cannot be calculated from data on the movement of point P1 alone unless information on the relationship between side P2 P2 and point P3 is given.

For simplicity of explanation only, the graphics object handling system according to the present invention is assumed to make and display segments and circles by selecting one or more control points from among the five sorts of points on the screen that are shown in Table 1.

TABLE 1

cp-type-1	ordinary point on the screen	independent point having no relationship with any existing object on the screen
cp-type-2	existing control point	control point such as vertex of an existing polygon, end of a line, or center of a circle
cp-type-3	intersection	intersection defined by an existing line, circle, and so forth
cp-type-4	internal division point	point dividing an existing line by a specific ratio
cp-type-5	point on a circumference	point located on a circumference and moved along it at a specific angle to a horizontal line on the screen

Referring now to Figure 11, an example of a graphics object handling system of the present invention includes a control unit (1), arithmetic unit (2), display control unit (3), display (4), mouse control unit (5), mouse (6), coordinate calculating unit (7), segment drawing unit (8), circle drawing unit (9), a memory arranged to provide graphics object table (10) and a control point table (11), and intersection point identifying unit (90). The control unit (1) performs overall control. The required calculation is done by the arithmetic unit (2). Control of the display (4) and mouse (6) is exerted, respectively, by the display control unit (3) and mouse control unit (5). Tables (10) and (11) are generated in a memory, and the reading of data to and from them is controlled directly by the control unit (1). Units (3), (5), (7), (8), (9), and (90) may be implemented by hardware for specific purposes, or in the form of software modules.

All graphics objects drawn on the screen of the display (4) are numbered and registered in the graphics object table (10). As shown in Table 2, the items for control points registered in the graphics object table (10) are "object number," "type," which determines the object to be a segment, circle, or something else, and "control point number" of one or more control points that define the object. Table 2 shows the cortents of the graphics object table (10) when a segment (object number: OBJ1) and a circle (object number: OBJ2) as shown in Figure 12 are displayed on the display screen.

BNSDOCID: <EP 0492901A2 >

TABLE 2

GRAPHICS OBJECT TABLE		
Object Number	Type	Control Point Number
....	....	....
OBJ1	Segment	CP1, CP2
....	....	....
OBJ2	Circle	CP3, CP4
....	....	....

The control point table (11) manages all control points that define graphics objects on the screen. Table 3 shows the contents of the control point table when the four control points CP1 to CP4 in Figure 12 are all cp-type-1, that is, ordinary points on the screen. Another example of a control point table is shown in Table 4. As shown below, the number and contents of the items for control points vary according to the types of the control points in Table 1. A process for generating Tables 3 and 4 is referred to later.

TABLE 3

CONTROL POINT TABLE (EXAMPLE 1)					
Control Point Number	Control Point Item 1 (type)	Control Point Item 2 (coordinate value)	Control Point Item 3	Control Point Item 4	Control Point Item 5
....	....	....	....	....	....
CP1	cp-type-1	(x1, y1)	....	....	....
CP2	cp-type-2	(x2, ys)	....	....	....
CP3	cp-type-3	(x3, y3)	....	....	....
CP4	cp-type-4	(x4, y4)	....	....	....
....	....	....	....	....	....

EPODOC: EP 0492901A2 1 >

TABLE 4

CONTROL POINT TABLE (EXAMPLE 2)					
Control Point Number	Control Point Item 1 (type)	Control Point Item 2 (coordinate value)	Control Point Item 3	Control Point Item 4	Control Point Item 5
....	....	....	....	....	....
CP11	cp-type-1	(x11, y11)	....	....	....
....	....	....	....	....	....
CP12	cp-type-2	(x11, y11)	CP11	....	....
....	....	....	....	....	....
CP13	cp-type-3	(x13, y13)	OBJ3	OBJ4	TO
....	....	....	....	....	....
CP14	cp-type-4	(x14, y14)	OBJ5	ratio1	....
....	....	....	....	....	....
CP15	cp-type-5	(x15, y15)	OBJ6	theta1	....
....	....	....	....	....	....

When a control point number is entered, the coordinate calculating unit (7) uses the number as a key to obtain from the control point table (11) a value indicating the type of the control point, and calculates and outputs the coordinate value of the control point on the screen according to its type. The operations of the unit are explained in detail later.

Figure 13 shows the screen of the display (4) when the graphics object handling system is activated. The operation screen may be designed in various ways; Figure 13 shows one example. Reference number (12) refers to a mouse cursor that moves in accordance with the movement of the mouse (6), a display area (13) which graphics object are displayed, a button area (14) for drawing segments, a button area (15) for drawing circles, and button areas (16) to (20) for designating the type of a control point.

The object generating function provided by the embodiment is explained below. First, an example in ~~which the graphics objects OBJ1 and OBJ2 shown in Figure 12 are generated and displayed in the display~~ area (13) is explained. Examples of procedures for generating and drawing these graphics objects according to the operation screen shown in Figure 13 are as follows:

(i) Procedure for drawing a segment CP1-CP2:

LS-1 When the mouse (12) is clicked on a button area (14), the system enters into the mode for drawing a segment.

LS-2 A temporary point (TP1) is displayed on the screen. The user moves it to a target position in the display area (13) by moving the mouse (6) while pressing the mouse button, a combined action which is hereinafter called dragging.

LS-3 When the mouse is clicked on a button area (16), the current coordinate value of TP1 is determined to be the coordinate value of a control point. The number CP1 is given to the control point. In this embodiment, a number indicating the setting order is assigned as a control point number.

LS-4 The value cp-tp-1, indicating that CP1 is an ordinary point on the screen, is registered as item 1 of the control point table for the control point number CP1, and the determined coordinate values are registered as item 2 (see Table 3).

LS-5 A temporary point (TP2) is displayed on the screen. The user moves it to a target position in the display area (13) by dragging it with the mouse. Meanwhile, the temporary segment CP1 TP2 may be displayed in the display area (13).

LS-6 When the mouse is clicked on a button area (16), the current coordinate values of TP2 are determined to be the coordinate values of the new control point. The new control point is assigned the control point number CP2.

LS-7 The value cp-type-1, indicating that CP2 is an ordinary point on the screen, is registered as item 1

BNSDOCID: <EP 0492901A2 I >

of the control point table for the control point number CP2, and the determined coordinate values are registered as item 2 (see Table 3). Generation of the segment is then completed. In this embodiment, a number indicating the order of generation is given to the graphics object.

LS-8 OBJ1 is registered in the "object number" column of the graphics object table (10), a value indicating the object is a segment is registered in the "type" column, and CP1 and CP2 are registered in the "control point number" column (see Table 2).

LS-9 The coordinate values of CP1 and CP2 are transferred to the segment drawing unit (8), and a segment whose ends are CP1 and CP2 is displayed in the display area (13).

(ii) Procedure for drawing a circle with point CP3 as its center and point CP4 on its circumference:

CS-1 When the mouse is clicked on a button area (15), the system enters into the mode for drawing a circle.

CS-2 A temporary point (TP1) is displayed on the screen. The user moves it to a target position in the display area by dragging it with the mouse.

CS-3 When the mouse is clicked on a button area (16), the current coordinate values of TP1 are determined to be the coordinate values of the new control point. Control point number CP3 is given to the new control point.

CS-4 The value cp-type-1, indicating that CP3 is an ordinary point on the screen, is registered as item 1 of the control point table for the control point number CP3 in the control point table, and the determined coordinate values are registered as item 2 (see Table 3).

CS-5 A temporary point (TP2) is displayed on the screen. The user moves it to a target position in the display area (13) by dragging it with the mouse. Meanwhile, a temporary circle having point CP3 as its center and point TP2 on its circumference may be displayed in the display area (13).

CS-6 When the mouse is clicked on the button area, the current coordinate value of TP2 is determined to be the coordinate value of the new control point. Control point number CP4 is given to the new control point.

CS-7 The value cp-type-1, indicating that CP4 is an ordinary point on the screen, is registered as item 1 of the control point table for the control point number CP4, and the determined coordinate values are registered as item 2 (see Table 3).

CS-8 OBJ2 is registered in the "object number" column of the graphics object table (10), a value indicating that it is a circle is registered in the "type" column, and CP3 and CP4 are registered in the "control point number" column (see Table 2).

CS-9 The coordinate values of CP3 and CP4 are transferred to the circle drawing unit (9), and a circle with CP3 as its center and CP4 on its circumference is displayed in the display area.

In the foregoing procedures for drawing a segment, a control point may be set in a geometric relation to an existing graphics object. This is performed as explained below. Steps LS-2, 3 and 4 are discussed as examples.

To set a control point on an existing control point.

LS-2-1. Referring to Figure 14, the user drags TP1 with the mouse and places it on an existing control point.

LS-3-1-1. The user clicks the mouse on a button area (17).

LS-3-1-2. An existing control point that has TP1 in its effective range is selected. Referring to Figure 15, the effective range of a control point is a rectangular area with the control point as its center, transverse dimension DX, and longitudinal dimension DY. In this embodiment, it is assumed that the effective ranges are constant in size for all control points. Therefore, no item for describing the effective range of a control point is needed in the control point table. If the user wants the effective range of the control point to be variable, an item for registering the effective ranges set by the user must be provided in the control point table. Here, assume that control point CP11 has been selected from the control point table. CP11 is a point defining a triangle T1.

LS-4-1. A number, CP12, is given to a new control point and the value cp-type-2 of its control point type is registered as item 1. The coordinate values (x11, y11) of the control point CP11 selected in LS-3-1-2 are registered in item 2 as the coordinate values of the control point CP12. The control point number CP11 is registered as item 3 (see Table 4).

After steps LS-5 to LS-9 have been executed, the segment L10 connecting point CP12 and point CP2 shown in Figure 16 is displayed.

To set a control point on a point of intersection of existing graphics objects.

LS-2-2. Referring to Figure 17, TP1 is dragged with the mouse and placed on a point of intersection of

BNSDOCID: <EP 0492901A2 1 >

existing graphics objects (see Figure 17).

LS-3-2-1. The user clicks the mouse button (18).

LS-3-2-2. For each possible pair of existing objects, the system checks whether the two objects forming the pair have an intersection point. If so, the coordinate values of the intersection point and the numbers of the two objects are registered in the intersection point table (Table 5). If the point is formed by an intersection of a circle and a segment, or of two circles the intersection point identifying unit (90), which will be explained later, is used, and the coordinate values and a characteristic value output from the unit are registered in the intersection point table.

TABLE 5

INTERSECTION POINT TABLE				
Intersection Number	Object Number	Object Number	Coordinates	Characteristic Value
....	....	....	....	....
1k-2	OBJkm-2(L)	OBJkn-2(L)	(xk-2, yk-2)	....
1k-1	OBJkm-1(L)	OBJkn-1(L)	(xk-1, yk-1)	....
1k	OBJkm (L)	OBJkn (C)	(xk, yk)	Tk
1k + 1	OBJkm + 1(C)	OBJkn + 1(L)	(xk + 1, yk + 1)	Tk + 1
1k + 2	OBJkm + 2(L)	OBJkn + 2(C)	(xk + 2, yk + 2)	Tk + 2
Note: (L) means a segment, and (C) means a circle.				

LS-3-2-3 An intersection point having TP1 in its effective range is selected from the intersection point table. Referring to Figure 18, the effective range of an intersection point is a rectangular area with the intersection point as its center, transverse dimension DX, and longitudinal dimension DY. In this embodiment, the effective ranges are assumed to be constant for all intersection points. Therefore, no item for describing the intersection point effective range is needed in the intersection table.

LS-3-2-4. The selected intersection point is determined to be a new control point. It is given a control point number CP13.

LS-4-2. The value cp-type-3 of the type of control point CP13 is registered as item 1. The coordinates ~~of the intersection point selected in LS-3-2-3 are registered as item 2. The object numbers (OBJ3 and~~ OBJ4) of the two objects defining the intersection point are registered as items 3 and 4. Further, in response to at least one of the two objects being a circle, the characteristic value TO of the intersection point is registered as item 5 (see Table 4). The characteristic value is known from the intersection table.

After steps LS-5 to LS-9 have been executed, the segment L11 connecting the points CP13 and CP2 shown in Figure 19 is displayed.

The intersection point identifying unit (90) will now be explained. Referring to Figure 20, a segment (L) and a circle (C), in general, have two intersection points. Two circles also generally have two intersection points. Assume that segment L1 has been drawn as a secondary graphics object by setting control point K1 upon one of the two intersection points in Figure 20 and control point K0 upon a point other than an intersection. Assume also that segment (L) and circle (C) have been moved as shown in Figure 21. In this case, unless it can be determined which of the two newly formed intersection points is the result of the movement of K1, segment L1 cannot be redrawn. However, the use of the intersection point identifying unit (90) makes it possible to identify and track two intersection points defined by a segment and a circle. Therefore, in Figure 21, it is possible to determine which of the two intersection points corresponds to K1, and to redraw segment L1.

The operations of the intersection point identifying unit (90) are as follows:

(I) Referring to Figure 22, in the fresh mode, when the coordinate values of two points defining a segment L and the equation of a circle (C) (the coordinate values of its center and its radius may be sufficient) are entered, the unit outputs the coordinate values of two intersection points and characteristic values corresponding to them.

(II) Referring to Figure 23, in the continuous mode, when the coordinate values of two points defining a

EPODOC ID: <EP 0492901A2 >

segment L, the equation of a circle (C) (the coordinates of its center and its radius alone may be sufficient), and a characteristic value are entered, the unit outputs the coordinate values of the intersection point corresponding to the characteristic value.

Described below is a method for distinguishing two solutions of simultaneous equations, that is, two intersection points, by using characteristic values. This method constitutes the fundamental portion of the operation. The embodiment discriminates two intersection points by using the sign of the inner product of vectors.

Referring to Figure 24, the ends of a segment (L) are named P and Q, and the points of intersection with a circle (C) K1 and K2. Vector is defined by P Q, and vector by K1 K2. A characteristic value t intrinsic to each intersection point is discussed. The characteristic value t may be 0 or 1. The inner product of vectors k = is calculated, and if k > 0, the characteristic value t of intersection point K1 is defined as 0, and the characteristic value t of intersection point K2 as 1. If k < 0, the characteristic value t of intersection point K1 is defined as 1, and the characteristic value t of intersection point K2 as 0. Since and are always parallel, examining the sign of their inner product is equivalent to examining whether the directions of the two vectors are the same or opposite. As can be seen from Figures 24 and 25, whichever of the two intersection points may be called K1, the characteristic value t of the intersection point nearer to P is defined as 0, and the characteristic value t of the intersection point nearer to Q as 1.

Even if the original graphics objects (L) and (C) are moved or reshaped, it is possible to distinguish which of two newly generated intersection points corresponds to which original intersection point, because one intersection point is thus identified by means of the characteristic value. Specifically, it is sufficient to solve the changed simultaneous equations for (L) and (C) and to select one of the two solutions that has the same characteristic value. Since this method merely examines the sign of an inner product of vectors, it is very efficient for identifying an intersection point.

The calculation of intersection points and definition of characteristic values are now described in more detail. If an xy coordinate system is set on a screen, a straight line (A) connecting two points P (xp, yp) and Q (xq, yq) is expressed by the following equation:

$$(yp - yq)*x + (xq - yp)*y + (xp*yq - xq*yp) = 0 \quad (F1)$$

A circle (C) having a point S(xs, ys) as its center and a radius r is expressed by the following equation:

$$(x - xs)**2 + (y - ys)**2 = r**2 \quad (F2)$$

where **2 indicates that the value is squared.

Referring to Figure 26, the values of xp, yp, xq, yq, xs, ys, and r are determined when the straight line and the circle are drawn. The intersection point identifying unit (90) solves simultaneous equations (F1) and (F2), and obtains x - y values, (x1, y1) and (x2, y2), that simultaneously satisfy the two equations. Referring to Figure 27, these two solutions are the coordinates of the intersection points K1 and K2 of straight line (A) and circle (C).

Two two-dimensional vectors and are now discussed. is the vector from point P to point Q on straight line (A), and is directed from P to Q. is the vector from K1 to K2, and is directed from K1 to K2. The components of and are expressed as follows:

= (xq - xp, yq - yp)
= (x2 - x1, y2 - y1)

The inner product k of and is:

$$k = * *\cos( )$$
$$= (xq - xp)*(x2 - x1) + (Yq - yp)*(y2 - y1) \quad \ldots (F3)$$

Since K1 and K2 are located on straight line (A), and are always parallel, and cos( ) always takes the value 1 or -1. Therefore, by calculating the value of inner product k from (F3) and examining its sign, the value of cos( ) can be determined. That is, it can be definitely determined whether the direction of is the same as or opposite to that of . The sign of k determines the characteristic value t, which takes the value 0 or 1. Referring to Figure 24, if k > 0, the characteristic value t of intersection point K1 is defined as 0, and

EPODOC: EP 0492901A2 1 >

the characteristic value t of intersection point K2 as 1. Referring to Figure 25, if k < 0, the characteristic value t of intersection point K1 is defined as 1, and the characteristic value t of intersection point K2 as 0.

In the fresh mode, the coordinates (x1, y1) and (x2, y2) of two intersection points and their characteristic values obtained in the foregoing process are output. In the continuous mode, the coordinates (x1, y1) and (x2, y2) of two intersection points and their characteristic values are obtained in the foregoing process, and the coordinates of whichever of these intersection points has the same characteristic value as the entered one are output.

The foregoing description has been directed to a case in which the intersection points are defined by a circle and a segment. This vector approach may be applied to identifying a point of intersection of two circles. That is, the embodiment distinguishes two intersection points by using the sign of one component of the outer product of vectors.

Referring to Figure 28, the centers of circles C1 and C2 are named S1 and S2, respectively, and the two intersection points of the circles K1 and K2, respectively. Vector is deFined by P Q, and vector by K1 K2. The plane encompassing the two vectors is assumed to be an xy plane. Further, the characteristic value t intrinsic to each intersection point is discussed. The characteristic value t may be 0 or 1. The outer product k = x is calculated. Referring to Figure 29, if its z component kz > 0, the characteristic value t of intersection point K1 is defined as 0, and the characteristic value t of intersection point K2 as 1. Referring to Figure 30, if kz < 0, the characteristic value t of intersection point K1 is defined as 1, and the characteristic value t of intersection point K2 as 0. Since and are always orthogonal, examining the sign of their outer product is equivalent to examining whether intersection point K1 or K2 is located to the right or left as viewed from the center S1. As can been seen in Figures 29 and 30, whichever of the two intersection points is called K1, the characteristic value t of the intersection point located to the right as viewed from center S1 is defined as 0, and the characteristic value t of the intersection point located to the left as 1. In this fashion, two intersection points defined by two circles can be identified.

The procedure for setting a control point in a geometric relation to an existing graphics object will now be further described.

To set a control point on an internal division point of an existing segment.

LS-2-3. Referring to Figure 30, TP1 is dragged with the mouse and placed on a target existing segment whose object number is OBJ5.

LS-3-3-1. When the mouse is clicked on a button area (19), the system enters the mode permitting the user to select a point on a segment using, for example, the meals disclosed in Japanese Patent Application 2-226812. The selected point is determined to be a new control point, and is given the number CP14.

LS-3-3-2. The ratio of internal division ratio 1 is calculated from the coordinates of the selected point and the coordinates of the ends of segment OBJ5. If the selected point is a midpoint, the resultant in ratio 1 is 0.5.

LS-4-3. The value cp-type-4 of the type of control point CP14 is registered as item 1. The coordinates of the internal division point are registered as item 2, the object number OBJ5 of the segment on which internal division point is located is registered as item 3, and the ratio of the internal division, ratio 1, is registered as item 4 (see Table 4).

Referring to Figure 32, after steps LS-5 to LS-9 have been executed, the segment L12 connecting points CP14 and CP2 is displayed.

To set a control point on the circumference of an existing circle.

LS-2-4. Referring to Figure 32, TP1 is dragged with the mouse and placed on the circumference of a target existing circle, whose object number is OBJ6.

LS-3-4-1 When the button (20) is clicked, the system enters the mode permitting the user to select a point on a circumference using, for example, the means described in Japanese Patent Application 2-226812. The selected point is determined to be a new control point, and is given the number CP15.

LS-3-4-2 An angle theta 1 that a segment connecting the point on the circumference and the center makes with a horizontal line on the screen is calculated from the coordinates of the selected point on the circumference, the coordinates of the center, and the radius of the circle.

LS-4-4 The value cp-type-5 of the type of control point CP15 is registered as item 1. The coordinates of the point on the circumference are registered as item 2, the object number OBJ6 of the circle is registered as item 3, and angle theta 1 is registered as item 4 (see Table 4).

Referring to Figure 33, after steps LS-5 to LS-9 have been executed, the segment L13 connecting points CP15 and CP2 is displayed.

In this fashion, a graphics object (secondary graphics object) is drawn by setting control points on the vertices of existing graphics objects (primary graphics objects), the points of intersection of theses objects, the internal division points of segments, points on circumferences, or other points. Note here that distinction

EPNSDOCID: <EP 0492901A2 >

between primary graphics objects and secondary graphics objects is relative. For example, it is possible to generate a graphics object OB2 that is geometrically constrained by graphics object OB1 and further to generate a graphics object OB3 that is geometrically constrained by graphics object OB2. In this case, OB2 is a secondary graphics object relative to OB1, but a primary object relative to OB3. Moreover, OB2 is a direct secondary graphics object of OB1, whereas OB3 is an indirect secondary graphics object of OB1.

A method for redrawing a secondary graphics object in response to the reshaping or movement of a primary graphics object will now be described. Control points other than cp-type-1 defining a secondary graphics object have data on their positions relative to other graphics objects in the control point table. That is, as shown in Table 4, the numbers of primary graphics objects, the numbers of their control points, and so forth are registered as the control point items. The coordinate calculating unit (7) calculates the current coordinate values of the control points by using such data. If a primary graphics object is changed in position or shape and if any secondary graphics object is geometrically constrained by the primary graphics object, the secondary graphics object is deleted from the screen, the values of the control points in the control point table are updated, and the secondary graphics object is displayed again.

Figure 34 shows the procedure by which the coordinate calculating unit (7) calculates the coordinates of control points according to their type. When a control point number is entered, the coordinate calculating unit (7) obtains, from the control point table, knowledge on the type of the control point that has the entered number, and calculates and outputs its current coordinate values by a method determined by its type.

CC-1 If the control point type is "ordinary point on the screen" (CP11 in Table 4):

The coordinate values in item 2 for the control point are unchanged.

CC-2 If the control point type is "existing control point" (CP12 in Table 4):

The "number of existing control points" in control point item 3 is entered again into the coordinate calculating unit, and its coordinate values are obtained. These values are output and also registered as control point item 2.

CC-3 If the control point type is "point of intersection existing graphics objects" (CP13 in Table 4):

The object numbers of the two intersecting graphics objects are registered as control point items 3 and 4. First, the two control points defining the graphics object that has the graphics object number in control point item 3 are obtained by consulting the graphics object table. Their current coordinate values (xp1, yp1) and (xp2, yp2) are calculated by using the coordinate calculating unit recursively. Similarly, the two control points defining a graphics object that has the graphics object number in control point item 4 are obtained, and their current coordinate values (xq1, yq1) and (xq2, yq2) are calculated. The different steps are executed in accordance with the types of the intersecting graphics objects.

i) If both the intersecting graphics objects are segments:

(xp1, yp1) and (xp2, yp2) are the coordinate values of the ends of one segment, and (xq1, yq1) and (xq2, yq2) are the coordinate values of the ends of the other. The solutions of the simultaneous equations given below are the coordinate values to be obtained. The obtained coordinate values are output and also registered as control point item 2.

$(yp1 - yp2)*x + (xp2 - xp1)*y + (xp1*yp2 - xp2*yp1) = 0$
... equation of a segment appearing in control point item 3
$(yq1 - yq2)*x + (xq2 - xq1)*y + (xq1*yq2 - xq2*yq1) = 0$
... equation of a segment appearing as control point item 4

(ii) If the intersecting graphics objects are a segment and a circle:

The graphics object appearing in control point item 3 is assumed to be a segment, whereas the graphics object appearing in control point item 4 is assumed to be a circle. (xp1, yp1) and (xp2, yp2) are the coordinate values of the ends of the segment, (xq1, yq1) are the coordinate values of the center of the

BNSDOCID: <EP 0492901A2 >

circle, and (xq2, yq2) are the coordinate values of a point on the circumference. The solutions of the simultaneous equations given below are the coordinate values to be obtained. The obtained coordinate values are output and also registered as control point item 2. At this time, a calculation and a decision on which of the two obtained pairs of coordinates should be chosen are carried out by the intersection point identifying unit (90). Specifically, the coordinate values of the two ends of the segment, the coordinate values of the center of the circle, the radius of the circle, and the characteristic value in control point item 5 corresponding to the intersection point are entered into the intersection point identifying unit (90), and the coordinate values to be obtained are then output.

(yp1 - yp2)*x + (xp2 - xp1)*y + (xp1*yp2 - xp2*yp1) = 0
... equation of a segment appearing in control point item 3
(x - xq1)**2 + (y - yq1)**2 = r**2
... equation of a circle appearing in control point item 4
where r = sqrt((xq2 - xq1)**2 + (yq2 - yq1)**2)

iii) If the intersecting graphics objects are a segment and a circle:

The procedure is the same as in (ii) except that the simultaneous equations of two circles are solved. In this case substantial processing is also executed by the intersection point identifying unit (90).

CC-4 If the control point type is "internal division point" (CP14 in Table 4):

The graphics object number of the segment to be divided is registered as control point item 3, and the ratio 1 of the internal division as control point item. The two control points defining the graphics object that has the graphics object number in control point item 3 are found by consulting the graphics object table. By using the coordinate calculating unit recursively, their recent coordinate values (xp1, yp1) and (xp2, yp2) are calculated. Using these, the current coordinate values of the internal division point are calculated from the equations given below, registered in item 2, and output.

x = xp1 + (xp2 - xp)*ratio1
y = yp1 + (yp2 - yp1)*ratio1

~~CC-5 If the control point type is "point on a circle" (CP15 in Table 4):~~

The graphics object number of the circle has been registered as control point item 3, and the value of angle theta 1 as control point item 4. The two control points defining a graphics object that has the graphics object number in control point item 3 are obtained by consulting the graphics object table. By using the coordinate calculating unit recursively, their current coordinate values (xp1, yp1) and (xp2, yp2) are calculated. Using these, the current coordinate values of the internal dividing point are calculated from the equations given below, registered in item 2, and output.

x = xp1 + r*cos(theta1)
y = yp1 + r*sin(theta1)

where r = sqrt((xp2 - xp1)**2 + (yp2 - yp1)**2), (xp1, yp1) are the coordinate values of the center of the circle, and (xp2, yp2) are the coordinate values of a point on the circumference.

The procedures for reshaping a primary object and for reshaping a secondary object in response are now explained. Objects OBJ1 and OBJ100 in Figure 35 are taken as an example. Table 6 shows a graphics object table corresponding to Figure 35, and Table 7 shows a corresponding control point table.

BNSDOCID: <EP_____0492901A2 I >

TABLE 6

GRAPHICS OBJECT TABLE		
Object Number	Type	Control Point Number
....	....	....
OBJ1	Segment	CP1, CP2
....	....	....
OBJ2	Circle	CP3, CP4
....	....	....
OBJ100	Segment	CP100, CP101

TABLE 7

CONTROL POINT TABLE					
Control Point Number	Control Point Item 1 (type)	Control Point Item 2 (coordinate value)	Control Point Item 3	Control Point Item 4	Control Point Item 5
....	....	....	....	....	....
CP1	cp-type-1	(x1, y1)	....	....	....
....	....	....	....	....	....
CP2	cp-type-1	(x2, y2)	....	....	....
....	....	....	....	....	....
CP3	cp-type-1	(x3, y3)	....	....	....
....	....	....	....	....	....
CP4	cp-type-1	(x4, y4)	....	—	—
....	....	....	....	....	....
CP100	cp-type-4	(x100, y100)	OBJ1	ratio100	....
....	....	....	....	....	....
CP101	cp-type-1	(x101, y101)	....	....	....
....	....	....	....	....	....

Given below is an explanation of a case in which point CP1 in Figure 35 is moved.

rs-1 When the user moves the mouse cursor to the vicinity of CP1 and presses the mouse button, one of the control points in the control point table, which includes the point designated by the mouse cursor in its control point effective range, is selected. The selected control point (CP1 in this case) is recognized as having been updated.

rs-2 When the user further moves the mouse cursor and presses the mouse button at a target position, the coordinate values (x1', y1') of the location currently designated by the mouse cursor are registered as item 2 of control point CP1.

rs-3 Referring to Figure 36, in response to the updating of the coordinate values of CP1 in the control point table, the redrawing process is started.

The redrawing process is started in response to a change in the coordinate values of at least one control point of cp-type-1, finds a graphics object at least one control point of which has had its coordinate values changed, and redraws this graphics object. In the redrawing process shown in Figure 36, the case in which N = OBJ2 is explained. (In Figure 36, N is the number of a graphics object, and n is the number of a control point that defines the graphics object.)

BNSDOCID: <EP 0492901A2 >

The graphics object table is first consulted, and it is found that OBJ2 is defined by control points CP3 and CP4 (ss-1). Next, information on the types of CP3 and CP4 is obtained from the control point table (ss-2). After this, it is determined whether CP3 and CP4 are cp-type-1. Since both CP3 and CP4 are cp-type-1 (ordinary points on the screen), they are not affected by a change of OBJ1. Therefore, since the circle (OBJ2) determined by these two points is not affected by such a change of OBJ1, step ss-4 and subsequent steps are skipped. That is, OBJ2 is not redrawn.

In the redrawing process, the case in which N is segment OBJ100 is explained. The graphics object table is first consulted, and it is found that OBJ100 is defined by control points CP100 and CP101 (ss-1). Next, information on the types of CP100 and CP101 is obtained from the control point table (ss-2). After this, it is decided whether CP100 and CP101 are cp-type-1. Here, CP100 is cp-type-4, and is a point geometrically related to the graphics object. Therefore, it is determined that it may be affected by a change in the position of graphics object OBJ1.

Here, since CP100 is a point directly related geometrically to graphics object OBJ1, it is certain that it will be affected by a change in the position of the latter. However, even if CP100 were a point directly related geometrically to another graphics object, it would still be possible for the graphics object to be a direct or indirect secondary graphics object relative to graphics object OBJ1. It should therefore be noted that, if the type of at least one control point is other then cp-type-1, the system proceeds to the next SS-4 to perform recalculation of the coordinate values of all control points.

In ss-4, the coordinate values of CP100 are calculated by the CC-4 function of the coordinate calculating unit by using geometric information on OBJ1 (i.e., coordinate data on CP1 and CP2) and ratio 100 (the value of the ratio). The coordinate values of CP101 are calculated by the CC-1 function of the coordinate calculating unit. Next, it is determined whether the coordinate values of any control point are changed. In the example, the coordinate values of CP100 have been changed to (x100', y100'). Therefore, item 2 for CP100 in the control point table is updated, and graphics object OBJ100 is then redrawn by using the updated coordinate value.

Referring to Figure 39, in the foregoing procedure, not only OBJ1, but also other graphics objects that have been generated under a geometrical constraint exerted directly or indirectly by OBJ1 can be redrawn so that they are affected by the change of OBJ1.

Referring to Figure 38, if the same procedure as for the redrawing process is followed, segment L10 is re-displayed in response to the reshaping of triangle T1 in Figure 16. Further, referring to Figure 39, segment L11 is re-displayed in response to the reshaping of the segment (OBJ3) in Figure 19, and, referring to Figure 40, segment L13 is re-displayed in response to the movement of the circle (OBJ6) in Figure 33 (Figure 40).

If the graphics object OBJ1 is translated, rotated, or scaled, the coordinate values of CP1 and CP2 in the control point table are updated simultaneously when a new position of object 1 is determined, and the coordinates of CP100 are then calculated and updated on the basis of the new coordinate values. The graphics object OBJ2 is reshaped to follow the movement of the graphics object OBJ1 by drawing a segment connecting CP100 and CP101, whose coordinates have been updated. Since a method for selecting a graphics object and editing it, such as translation, is known, its explanation is omitted.

Referring to Figure 34, some variations may be employed for the coordinate calculating unit and also referring to Figure 36, the redrawing process. For example, a step for determining whether the type of the control point is cp-type-1 may be added at the beginning of step ss-4, and coordinate calculations for cp-type-1 control points may be skipped. In this case, the coordinate calculating unit need not have the function cc-1.

Up to this point, the present invention has been explained by using an embodiment for making and redrawing segments and circles. But the scope of the present invention is not limited to these operations. Since straight lines or half-lines can be defined by two points on them, they can be generated and redrawn in the same way as a segment. Since points, polygons, quadratic curves other than circles, and three-dimensional solids can also be defined by using control points, the present invention can be applied to them. For instance, in the case of a point, a graphics object is defined by a single control point. According to the invention, if the control point for a point graphic object is set on the midpoint of an existing segment, the segment is a primary graphic object and the point is a secondary graphic object. By describing the relationship between the segment and the control point in the control point table, the point can always be constrained to the midpoint of the segment. A plane in three-dimensional space in turn is defined by three points. According to the present invention, if control points for the plane are set on the vertices of three cones and the relationships between the control points and the cones are described in the control point table, the plane can be redrawn in response to the movement of one or more cones.

Note also that the types of control points are not limited to those shown in Table 1. All the points in

EPNSDOCID: <EP 0492901A2 >

Table 1 except ordinary points on the screen are points located on existing graphics objects. A control point may be, although not located on an existing graphics object, geometrically related to that graphics object. One example of such a point is an external division point of a segment.

Furthermore, a unit for displaying a graphics object may be an apparatus other than a display, such as a printer. It is most preferable, however, to instruct an object to be reshaped or moved by using a mouse cursor on a display screen.

Note that the present invention is applicable regardless of the algorithm used for displaying graphics objects. A curve is generally approximated in a display by a continuous series of segments (see, for example, JA PUPA 2-250178). For this purpose, a table is provided for storing data on a continuous series of segments representing a graphics object.

It will now be appreciated that, according to the invention, the data held on a control point geometrically constrained by a graphics object are not resticted to the coordinates of the point at which it is set.

Further, according to the present invention, even when a number of graphics objects are interrelated, in response to the movement or reshaping of a graphics object, another graphics object generated under a geometical constraint exerted directly or indirectly by it can be redrawn while maintaining the constraint relationship between them. In addition, such redrawing can be done in response to a movement or reshaping of a graphics object other than translation, rotation, and scaling.

Claims

1. Apparatus for generating a graphics object by setting one or more control points for defining said object, comprising:
   (a) a graphics object table that stores, for each graphics object, an identifier thereof and identifiers of one or more control points;
   (b) a control point table that stores, for each control point, an identifier and positional data thereof, said positional data being indicative of the position of said control point relative to a graphics object; and
   (c) means of updating said graphics object table and said control point table in response to the setting of one or more control points for a new graphics object by a user.

2. Apparatus as claimed in claim 1 including means for displaying a graphics object that has been or is being generated.

3. Apparatus as claimed in claims 1 and 2 including means for helping a user to set a control point for a graphics object to be newly generated upon a control point of a previously generated graphics object, said means (c) being to register, in said control point table, the identifier of the control point that has been set and the identifier of the control point for said previously generated graphics object, in response to the setting of said control point by the user.

4. Apparatus as claimed in claims 1 and 2 including means for helping a user to set a control point for a graphics object to be newly generated upon an intersection point defined by two previously generated graphics objects, said means (c) being to register the identifier of the control point that has been set and the identifiers of the graphics objects defining said intersection point in response to the setting of said control point by the user.

5. Apparatus as claimed in claim 4 including a means for generating identification data for said intersection point if said two previously generated graphics objects are a quadratic curve and a line graphics object, or two quadratic curves, said means (c) registering said generated data in said control point table.

6. Apparatus as claimed in claims 1 and 2 including means for helping a user to set a control point for a graphics object to be newly generated upon an internal division point of a segment that is a previously generated graphics object, and means for calculating the ratio of the internal division of the segment, said means (c) being to register, in said control point table, the identifier of the control point that has been set, the identifier of the previously generated graphics object, and the ratio of the internal division, in response to the setting of said control point by the user.

7. Apparatus as claimed in claims 1 and 2 including means for helping a user to set a control point for a

EP 0 492 901 A2

graphics object to be newly generated upon the circumference of a circle that is a previously generated graphics object, and means for calculating the angle that a segment connecting the point on the circumference and the center makes with a reference line, said means (c) being to register, in said control point table, the identifier of the control point that has been set, the identifier of the previously generated graphics object, and said angle, in response to the setting of said control point by the user.

8. Apparatus for redrawing a graphics object generated under a geometrical constraint exerted directly or indirectly by another graphics object in response to the reshaping or movement of the latter, comprising:

(a) a graphics object table that stores, for each graphics object, an identifier thereof and one or more control points defining said graphics object;

(b) a control point table that stores, for each of said control points, an identifier and positional data thereof, said positional data being permitted to be data on the position of said control point relative to a graphics object;

(c) means of referring to said graphics object table and said control point table, in response to the reshaping or movement of a graphics object, in order to extract the identifiers of graphics objects that may have been generated under a geometrical constraint exerted directly or indirectly by the graphics object;

(d) means for calculating the coordinate values of control points which are for the extracted graphics objects and whose positional data are related to graphics objects, by utilizing said relative positional data; and

(e) means for displaying a graphics object defined by a control point at least one calculated coordinate value of which is different from the original value.

9. Apparatus as claimed in claim 8 including means for updating said control point table in accordance with the calculation results by said means (d).

10. Apparatus as claimed in claim 8 wherein said means (c) extracts the identifier of a graphics object if at least one control point defining the graphics object has data on its own position relative to another graphics object.

11. Apparatus as claimed in claims 8 to 10 wherein said data on the relative position of said control point consist of the identifier of the control point upon which said control point has been set.

12. ~~Apparatus as claimed in claims 8 to 10 wherein said data on the relative position of said control point~~ consist of the identifiers of two graphics objects defining the intersection point upon which said control point has been set.

13. Apparatus as claimed in claim 12 wherein, when said two graphics objects are a quadratic curve and a line graphic object, or two quadratic curves, identification data for said intersection point are included in said data on the relative position of said control point.

14. Apparatus as claimed in claims 8 to 10 wherein said data on the relative position of said control point consist of the identifier of a segment object and an internal division ratio that defines an internal division point of the segment upon which said control point has been set.

15. Apparatus as claimed in claims 8 to 10 wherein said data on the relative position of said control point consist of the identifier of a circular object and an angle that defines the point on its circumference upon which said control point has been set.

16. A method of generating a graphics object by setting one or more control points to define the graphics object comprising:

(a) making a graphics object table for storing, for each graphics object, an identifier thereof and identifiers of one or more control points;

(b) making a control point table that stores, for each control point, an identifier and positional data thereof, said positional data on the position being permitted to be data on the position of said control point relative to a graphics object; and

(c) updating said graphics object and said control point table in response to the setting of one or

BNSDOCID: <EP 0492901A2 I >

more control points for a new graphics object by a user.

17. In a graphics object handling system having a graphics object table that stores, for each graphics object, an identifier thereof and identifiers of one or more points defining the graphics object (hereafter called control points) and a control point table that stores, for each said control point, an identifier and positional data thereof, said positional data being permitted to be data on the position of the control point relative to a graphics object,

a method of redrawing a graphics object generated under a geometrical constraint exerted directly or indirectly by another graphics object in response to the reshaping or movement of the latter, comprising:

(a) in response to the reshaping or movement of a graphics object, referring to said graphics object table and said control point table to extract the identifiers of graphics objects that may have been generated under a geometrical constraint exerted directly or indirectly by the graphics object;

(b) calculating the coordinate values of control points which are for the extracted graphics objects and whose positional data are related to graphics objects, by utilizing said relative positional data; and

(c) displaying a graphics object defined by a control point at least one calculated coordinate value of which is different from the original value.

EPODOC: EP 0492901A2 1 >

FIG. 1

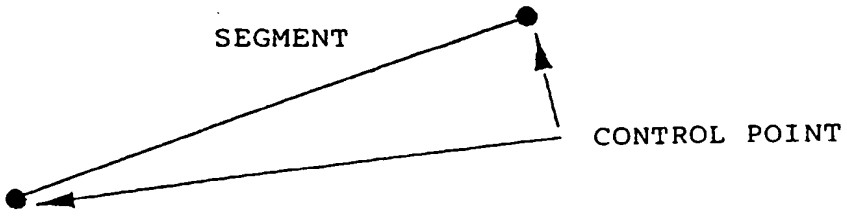
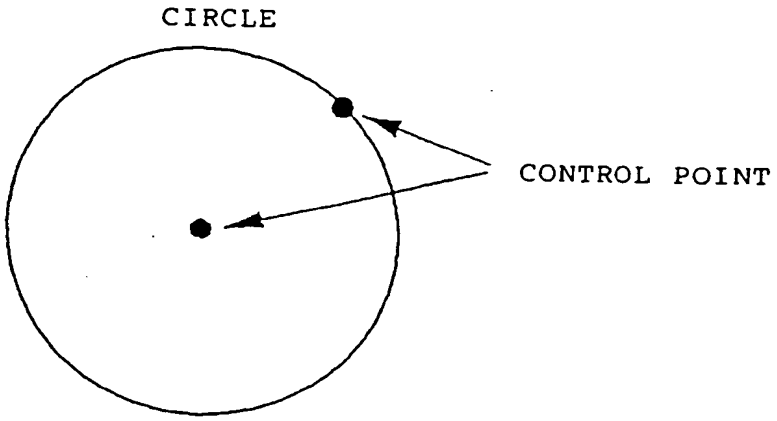


FIG. 2



BNSDOCID: <EP_____ 0492901A2 I >

FIG. 3

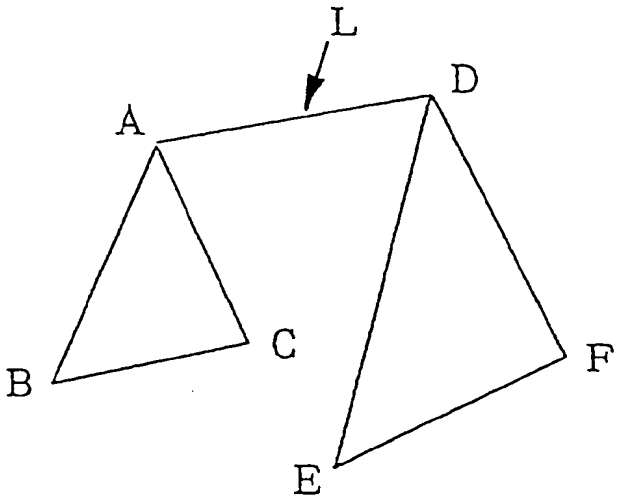
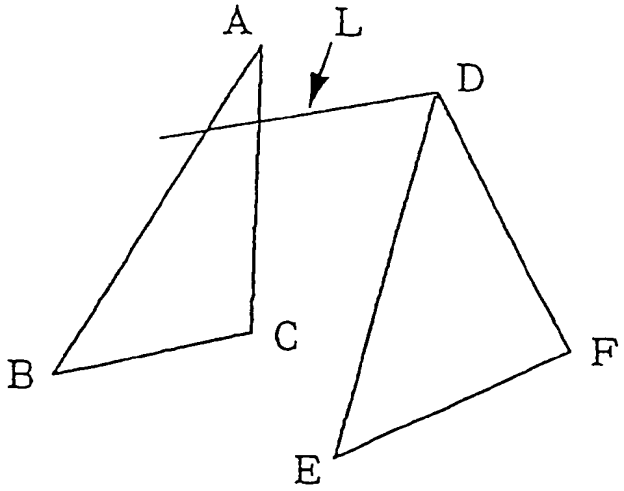


FIG. 4



PRIOR ART

EPODOC: EP 0492901A2 1 >

FIG. 5

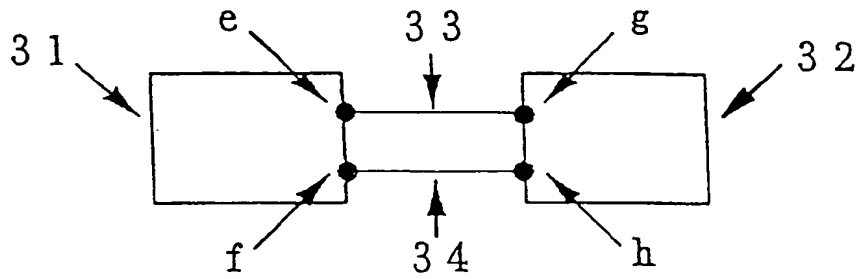


FIG. 6

GRAPHICS OBJECT ID	CONNECTED GRAPHICS OBJECT ID
31	33, 34
32	33, 34
33	31, 32
34	31, 32

PRIOR ART

FIG. 7

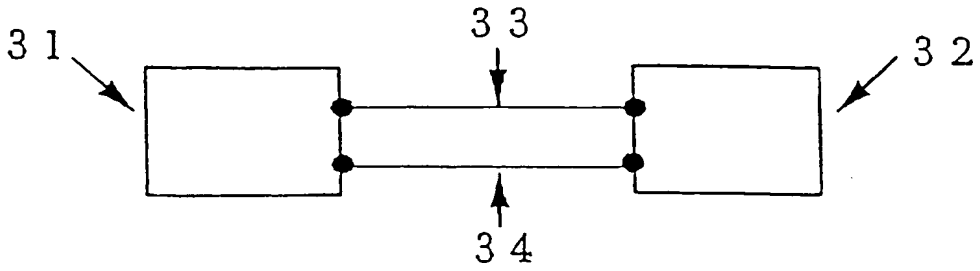


FIG. 8

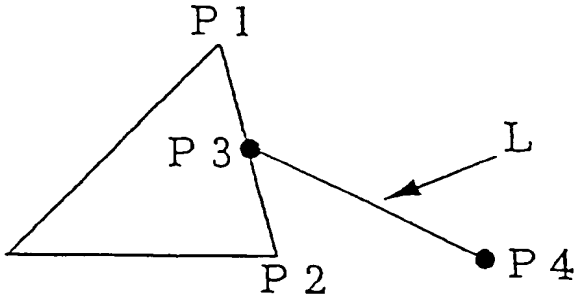


FIG. 9

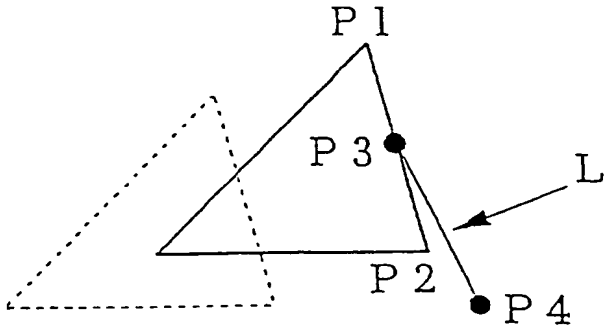
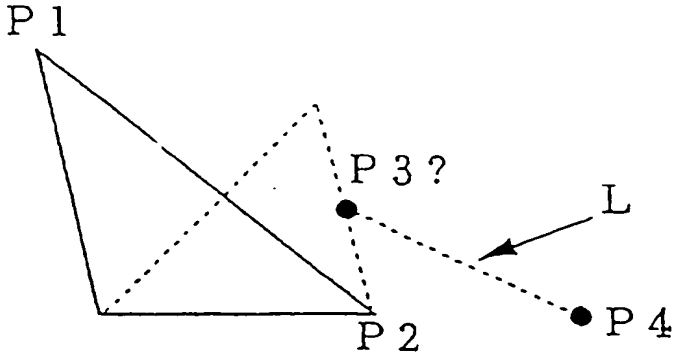
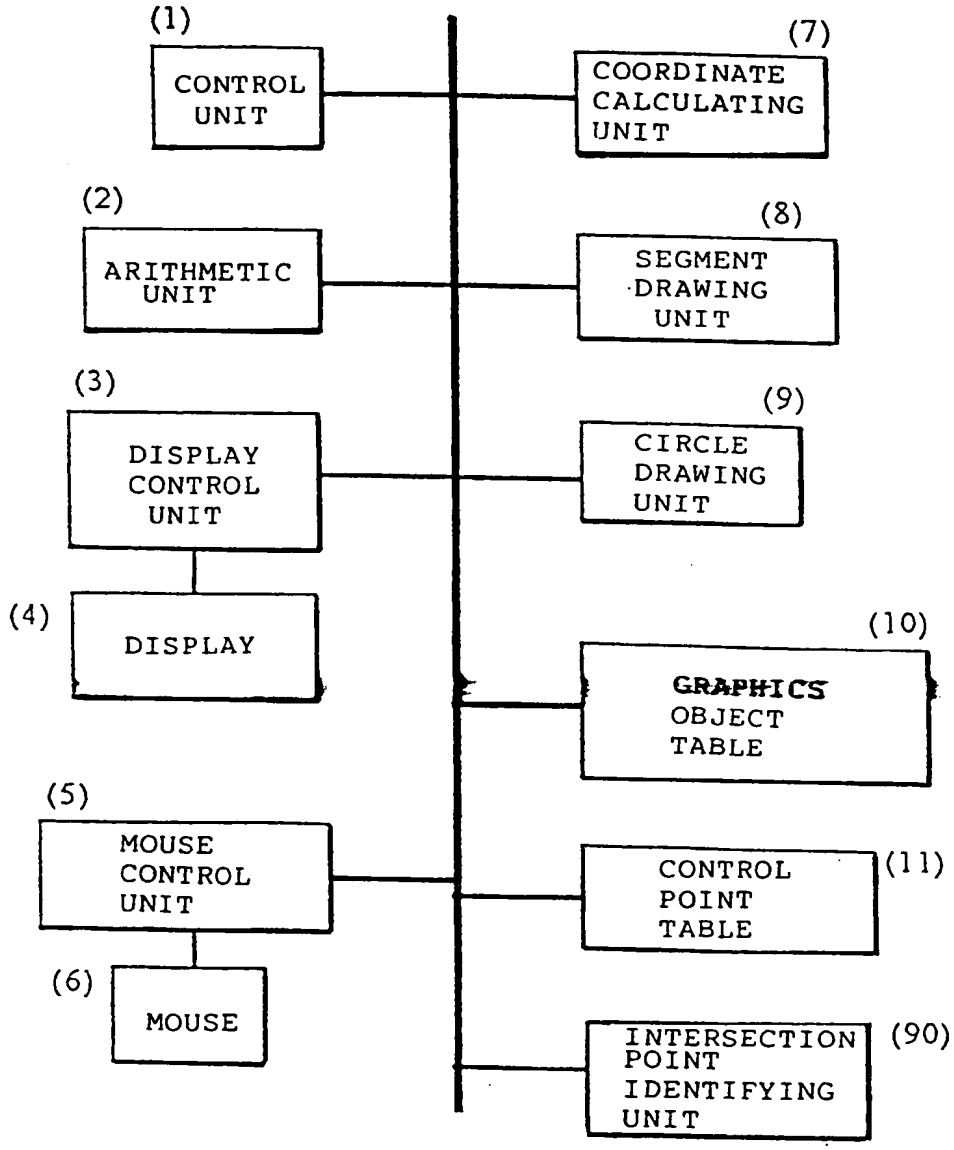


FIG. 10



PRIOR ART

FIG. 11



BNSDOCID: <EP 0492901A2 >

FIG. 12

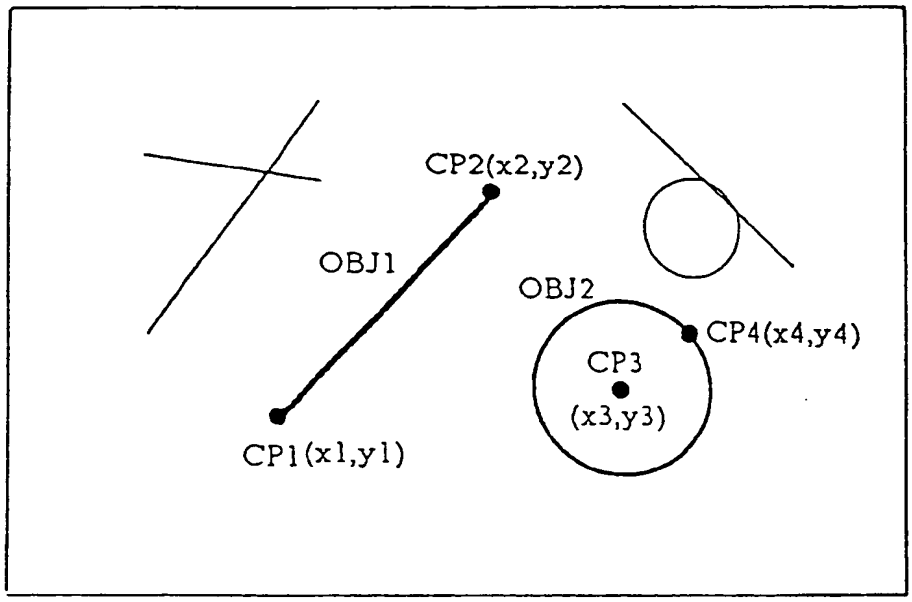

CP2(x2,y2)
OBJ1
OBJ2
CP4(x4,y4)
CP3
(x3,y3)
CP1(x1,y1)

FIG. 13

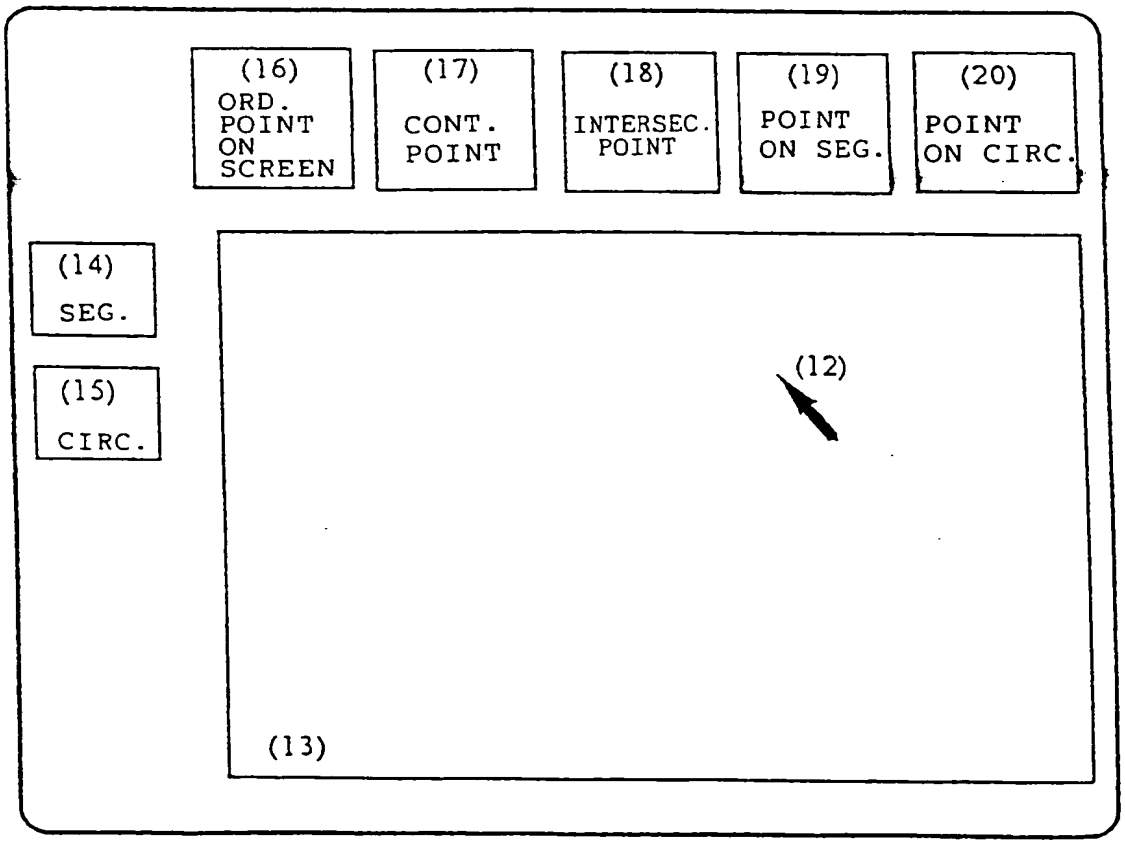

(16) ORD. POINT ON SCREEN
(17) CONT. POINT
(18) INTERSEC. POINT
(19) POINT ON SEG.
(20) POINT ON CIRC.
(14) SEG.
(15) CIRC.
(12)
(13)

FIG. 14

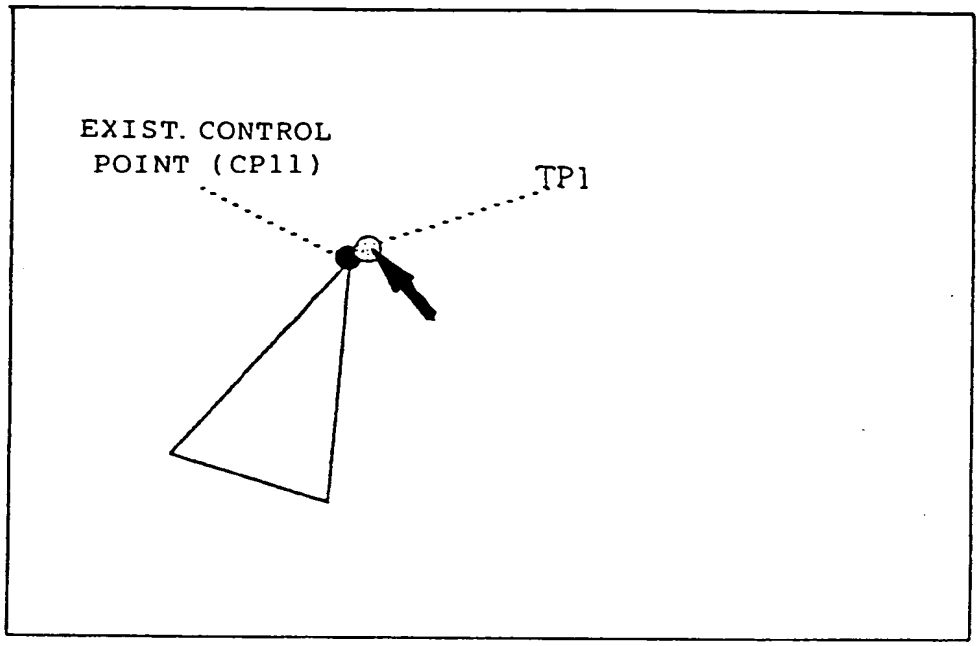
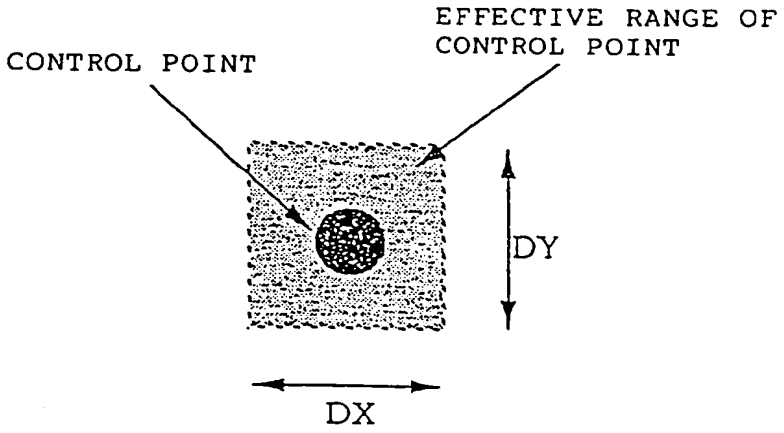


FIG. 15



EPODOC: <EP 0492901A2 >

FIG. 16

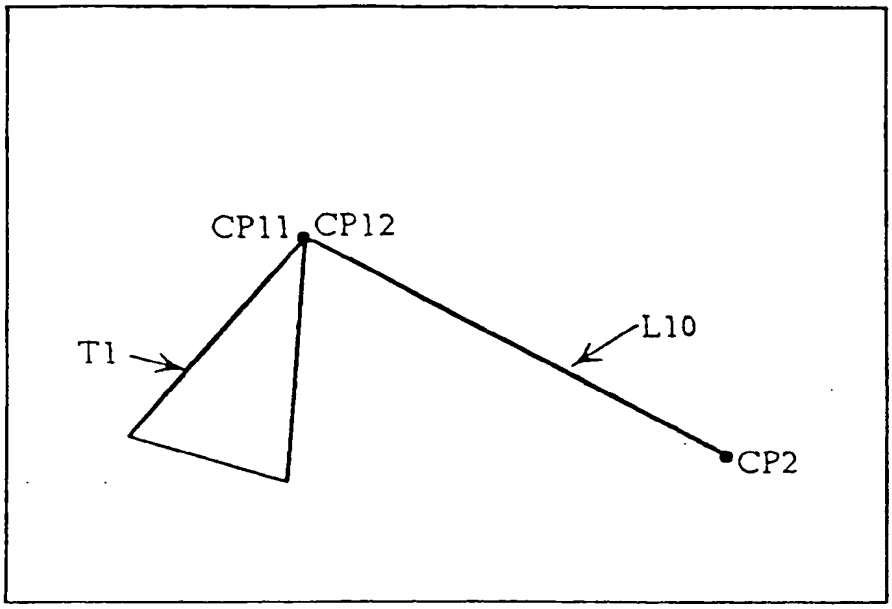


FIG. 38

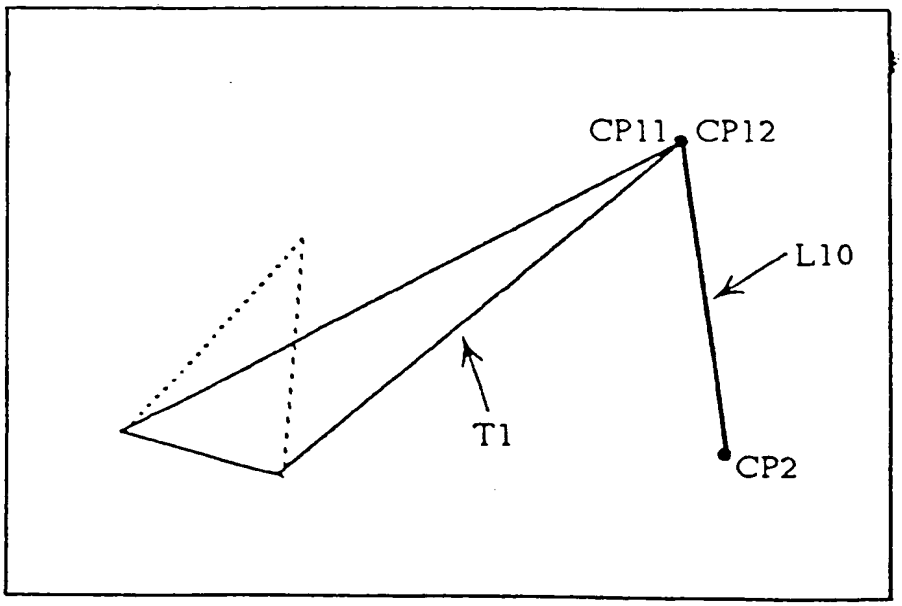


FIG. 17

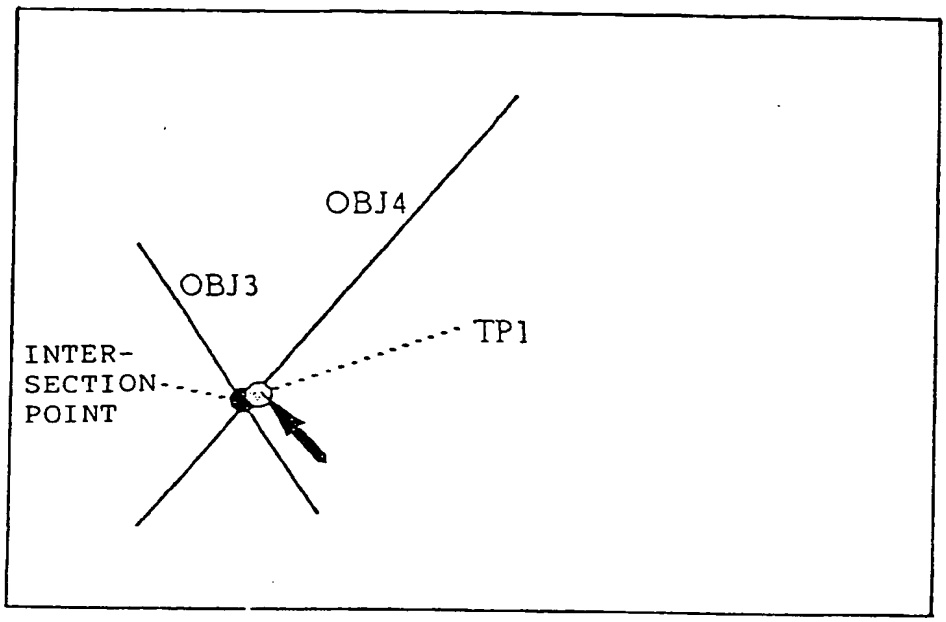
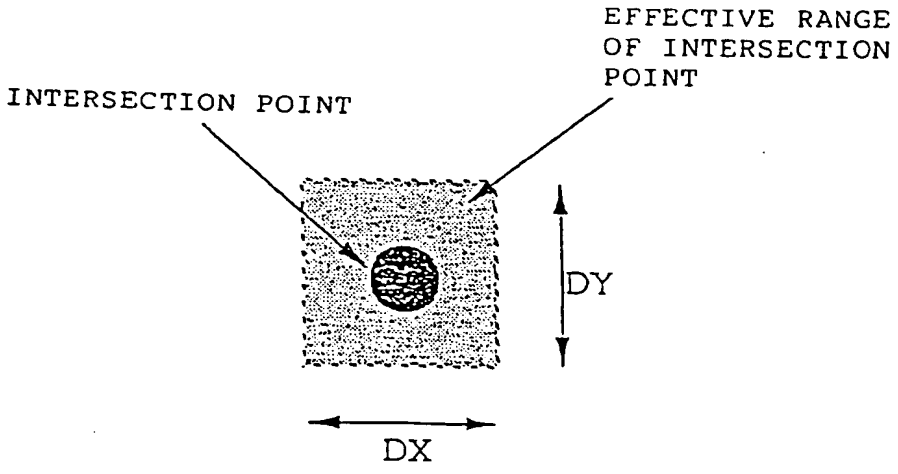


FIG. 18



BNSDOCID: <EP_____0492901A2_I_>

FIG. 19

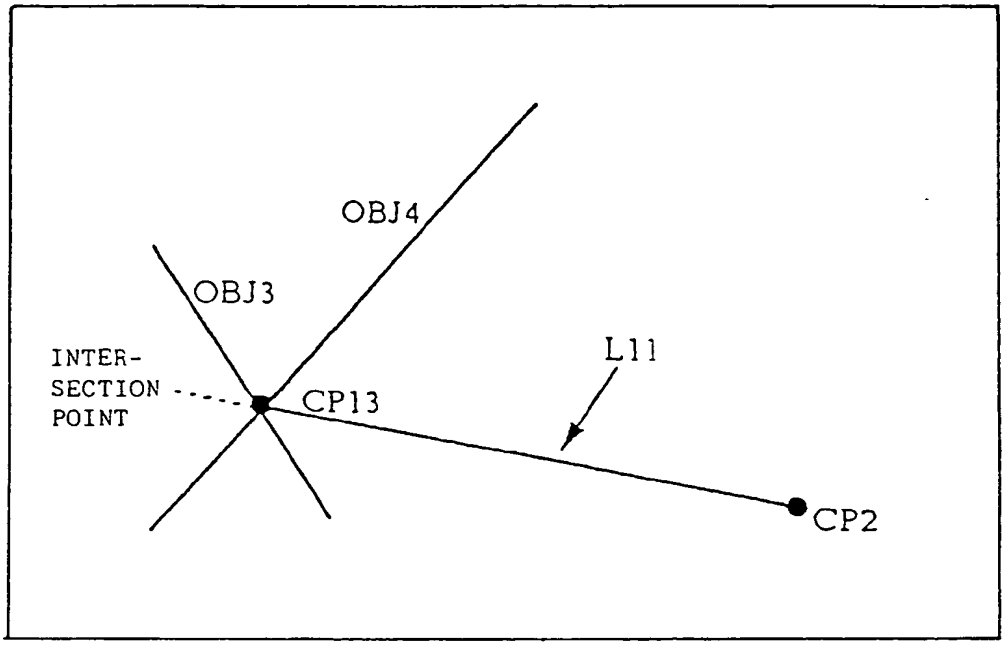

OBJ4
OBJ3
INTER-
SECTION
POINT
CP13
L11
CP2

FIG. 39

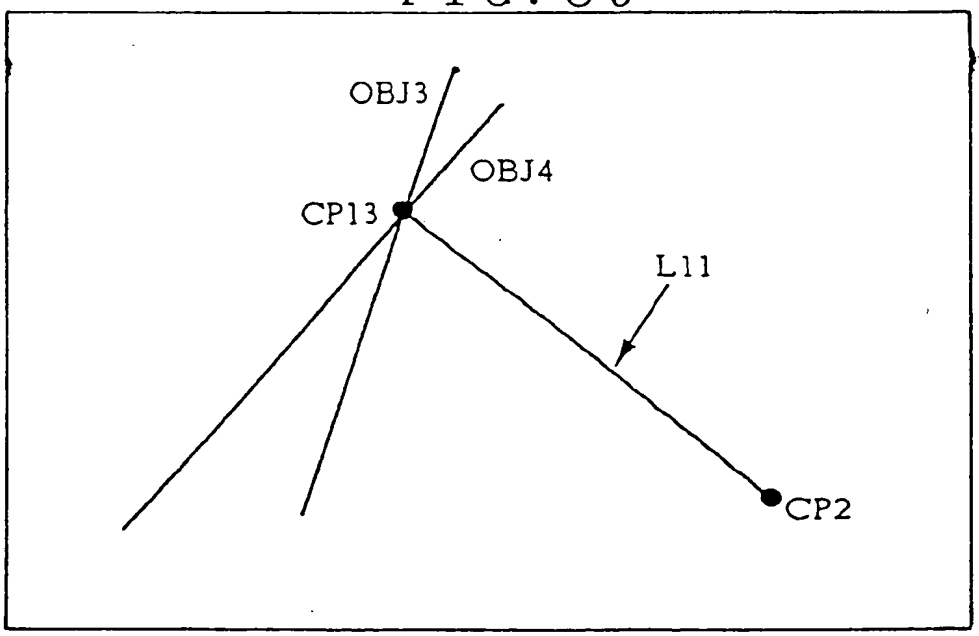

OBJ3
OBJ4
CP13
L11
CP2

FIG. 20

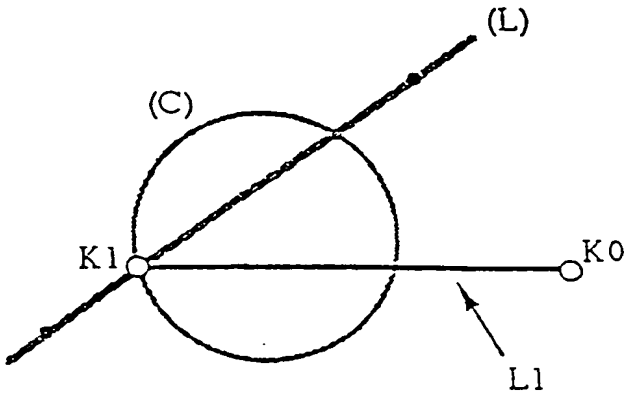
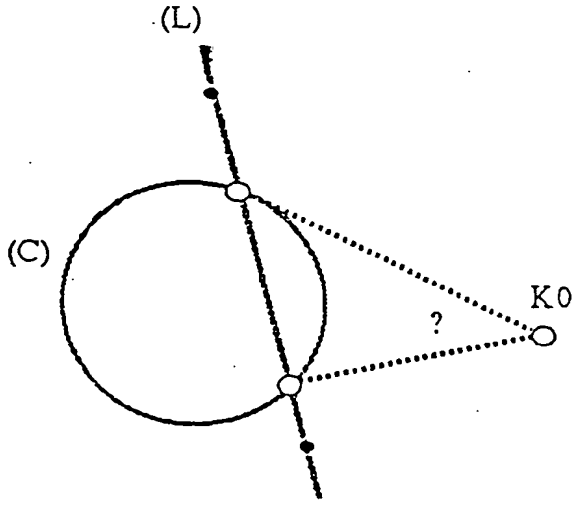


FIG. 21



EPNSDOCID: <EP 0492901A2 1 >

FIG. 22

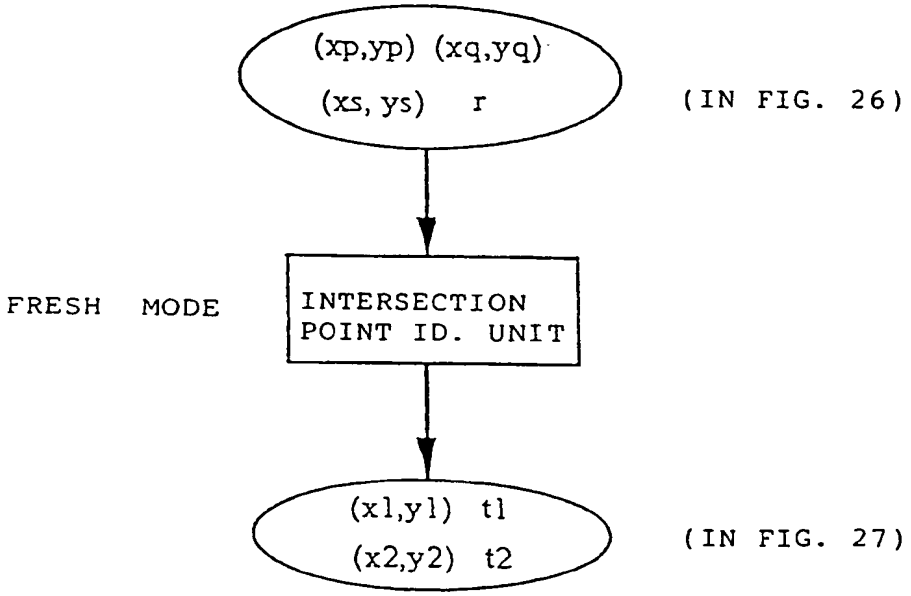

(xp,yp) (xq,yq)
(xs, ys) r
(IN FIG. 26)
FRESH MODE
INTERSECTION POINT ID. UNIT
(x1,y1) t1
(x2,y2) t2
(IN FIG. 27)

FIG. 23

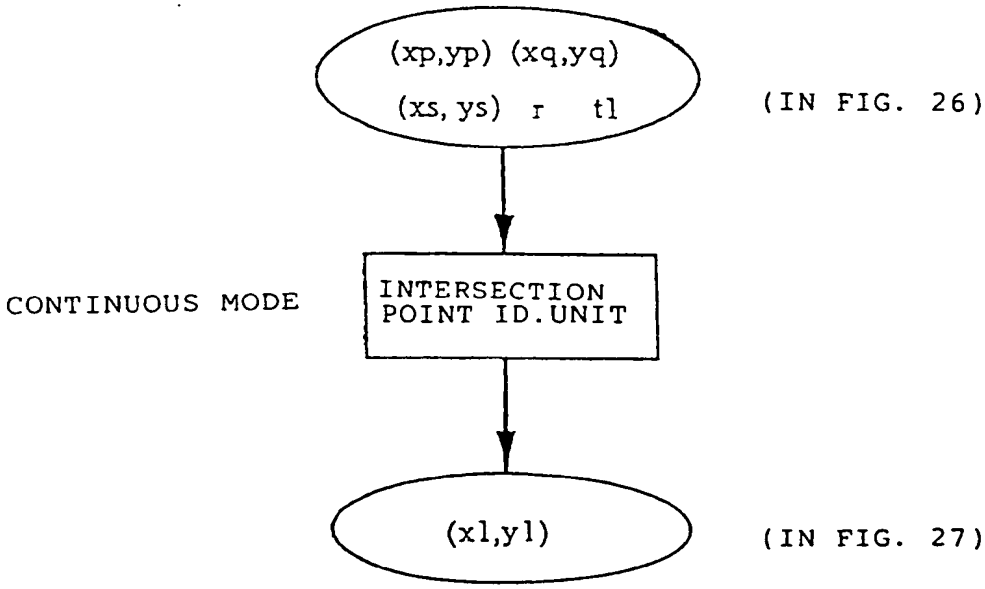

(xp,yp) (xq,yq)
(xs, ys) r t1
(IN FIG. 26)
CONTINUOUS MODE
INTERSECTION POINT ID.UNIT
(x1,y1)
(IN FIG. 27)

FIG. 24

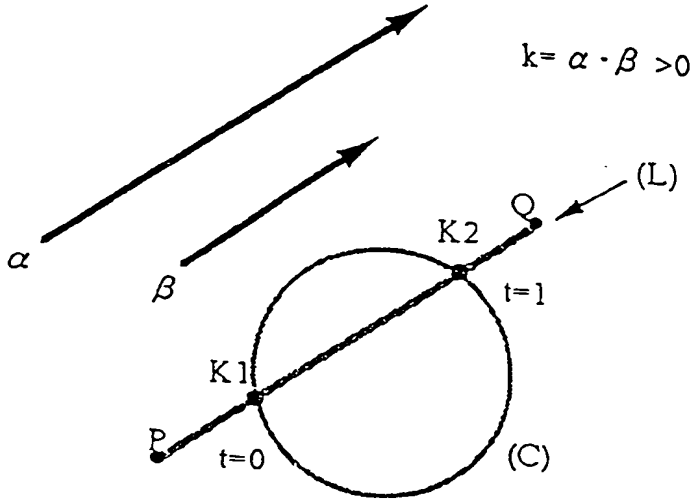
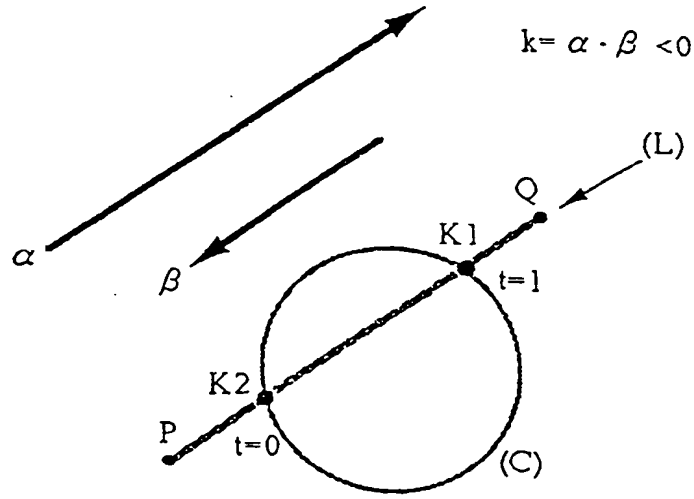


FIG. 25



BNSDOCID: <EP 0492901A2 I >

FIG. 26

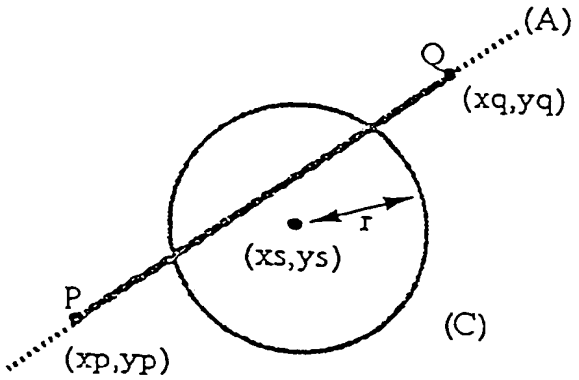

(A)
Q
(xq,yq)
r
(xs,ys)
P
(C)
(xp,yp)

FIG. 27

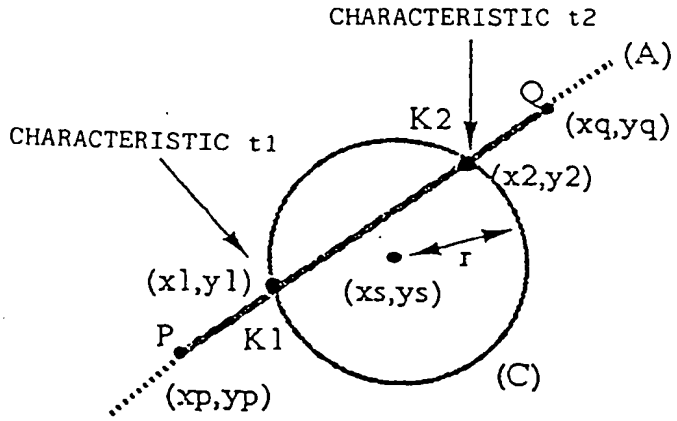

CHARACTERISTIC t2
(A)
Q
K2
(xq,yq)
CHARACTERISTIC t1
(x2,y2)
r
(x1,y1)
(xs,ys)
P
K1
(C)
(xp,yp)

FIG. 28

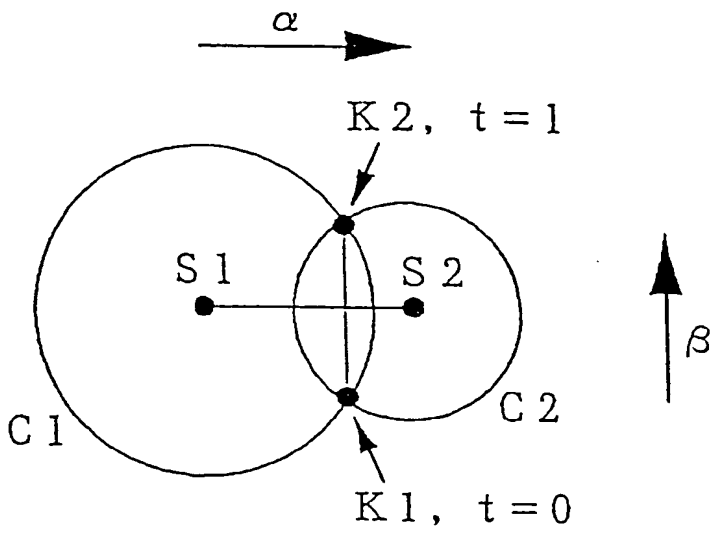
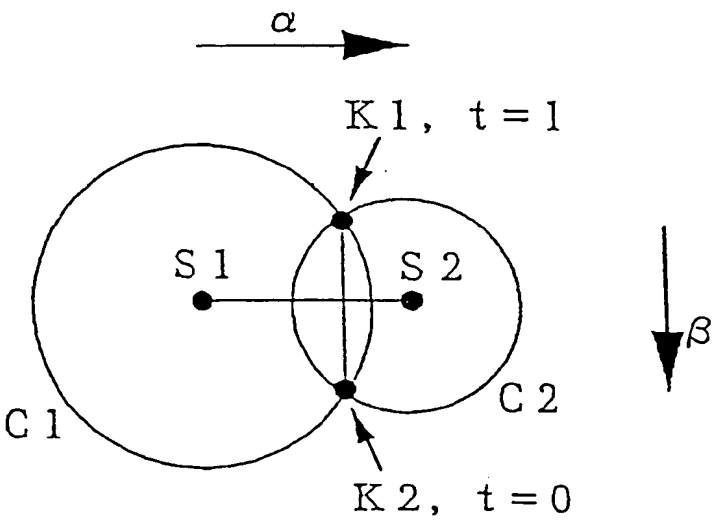


FIG. 29



BNSDOCID: <EP 0492901A2 I >

FIG. 30

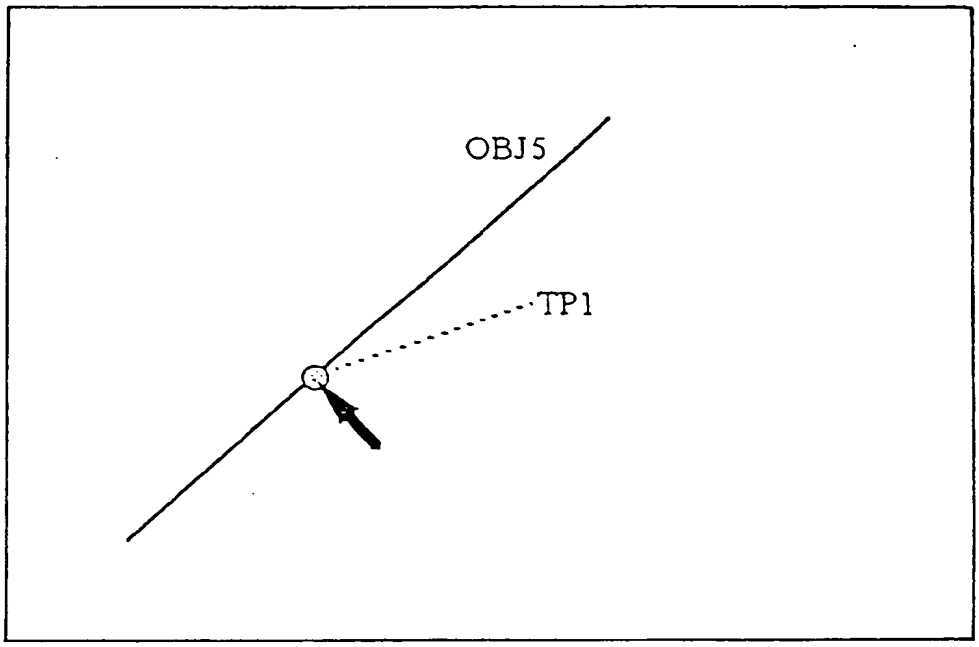


FIG. 31

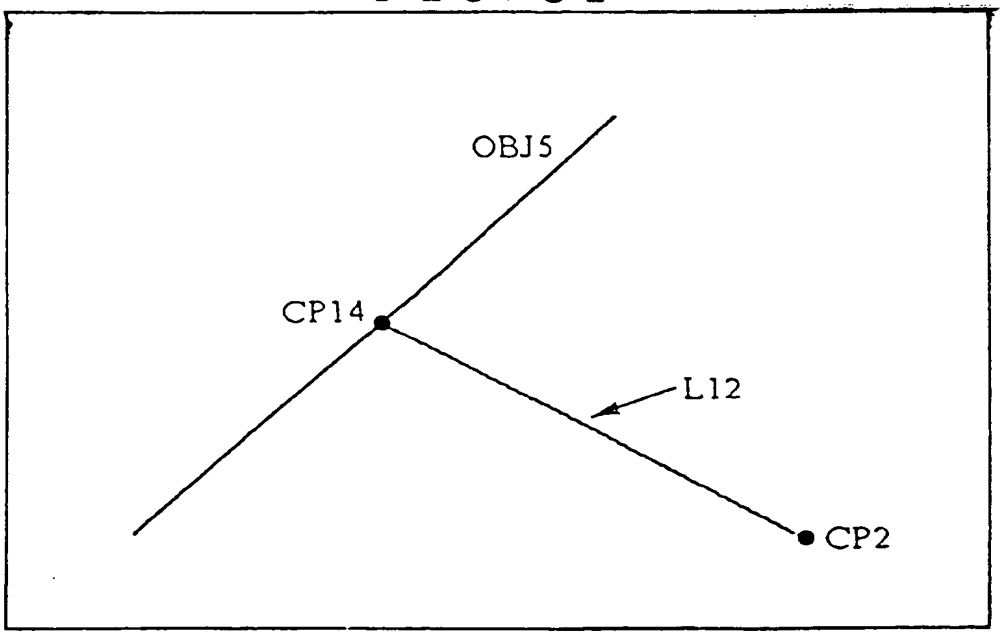


FIG. 32

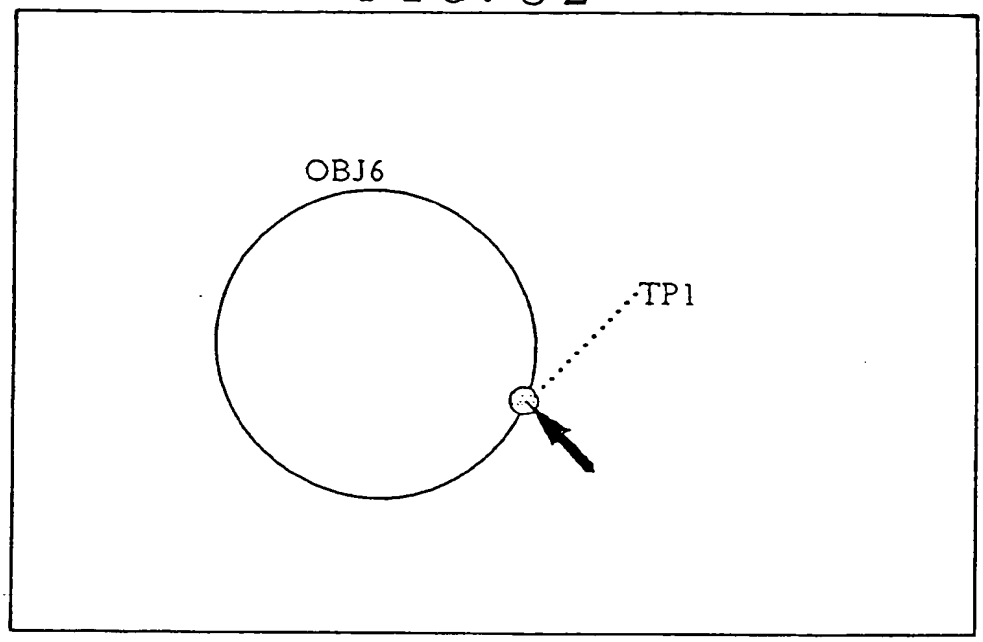

OBJ6
TP1

FIG. 35

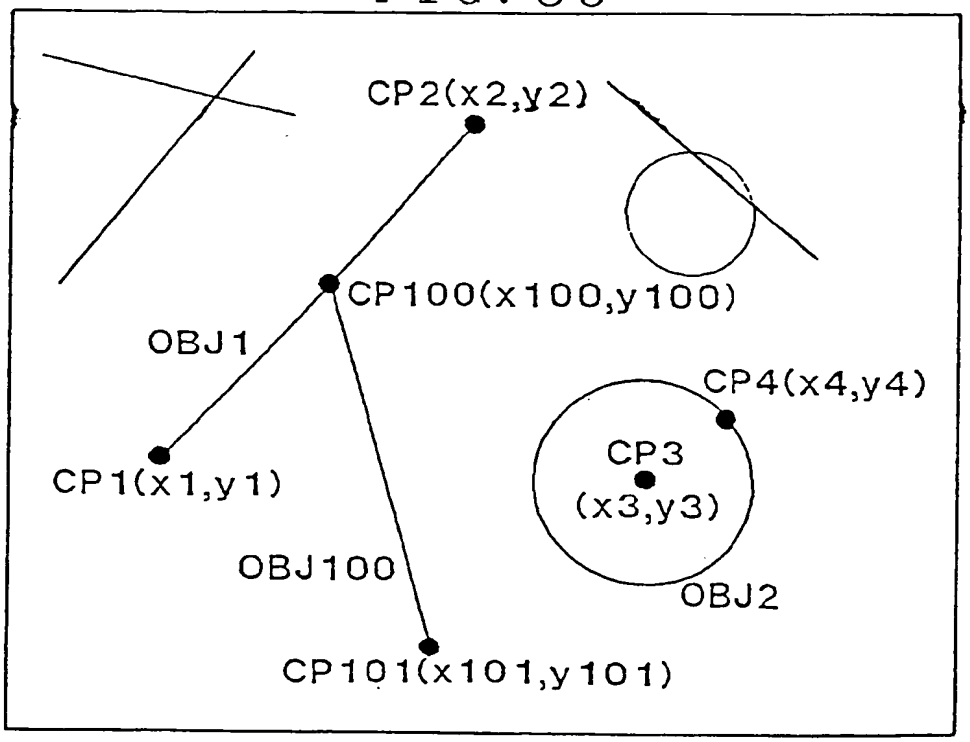

CP2(x2,y2)
CP100(x100,y100)
OBJ1
CP4(x4,y4)
CP3
(x3,y3)
CP1(x1,y1)
OBJ100
OBJ2
CP101(x101,y101)

FIG. 33

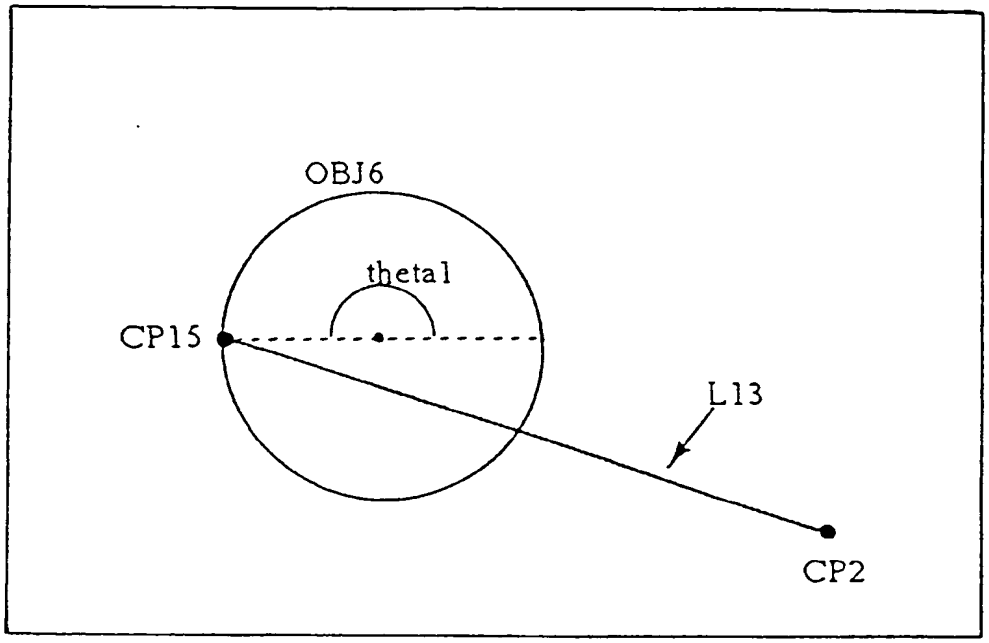


FIG. 40

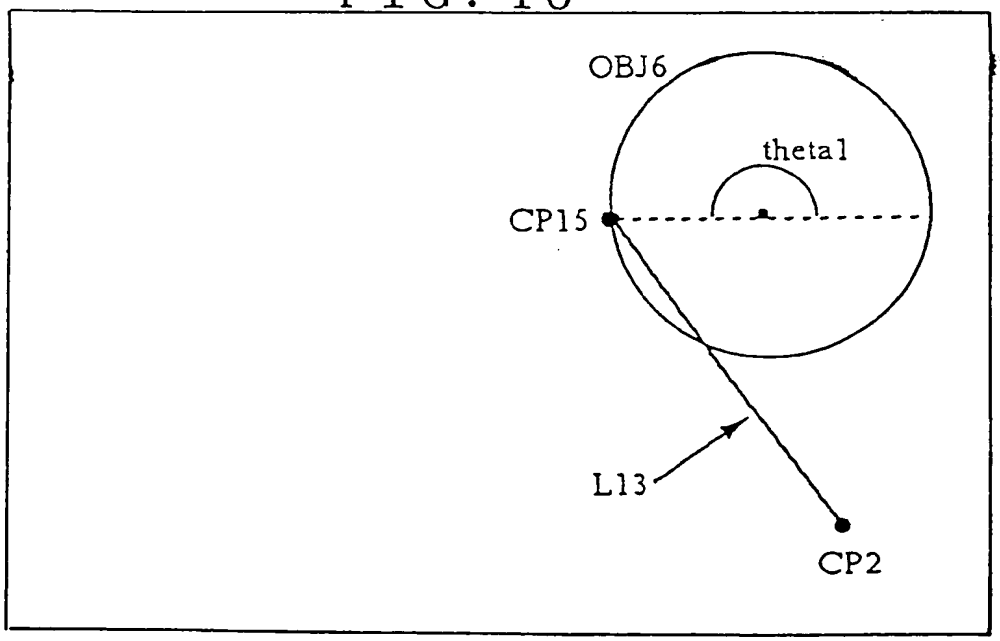
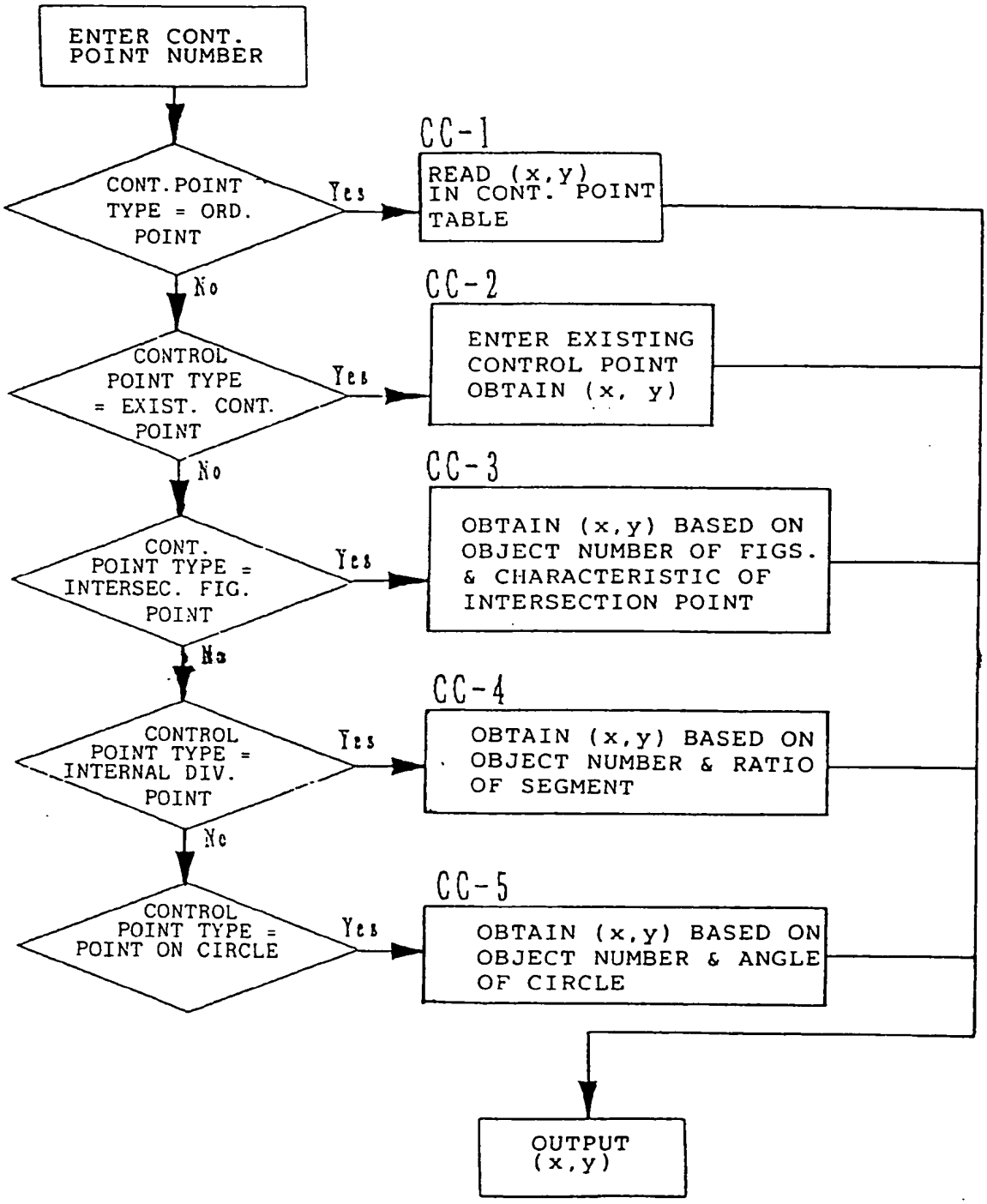


FIG. 34



BNSDOCID: <EP 0492901A2 >

FIG. 36

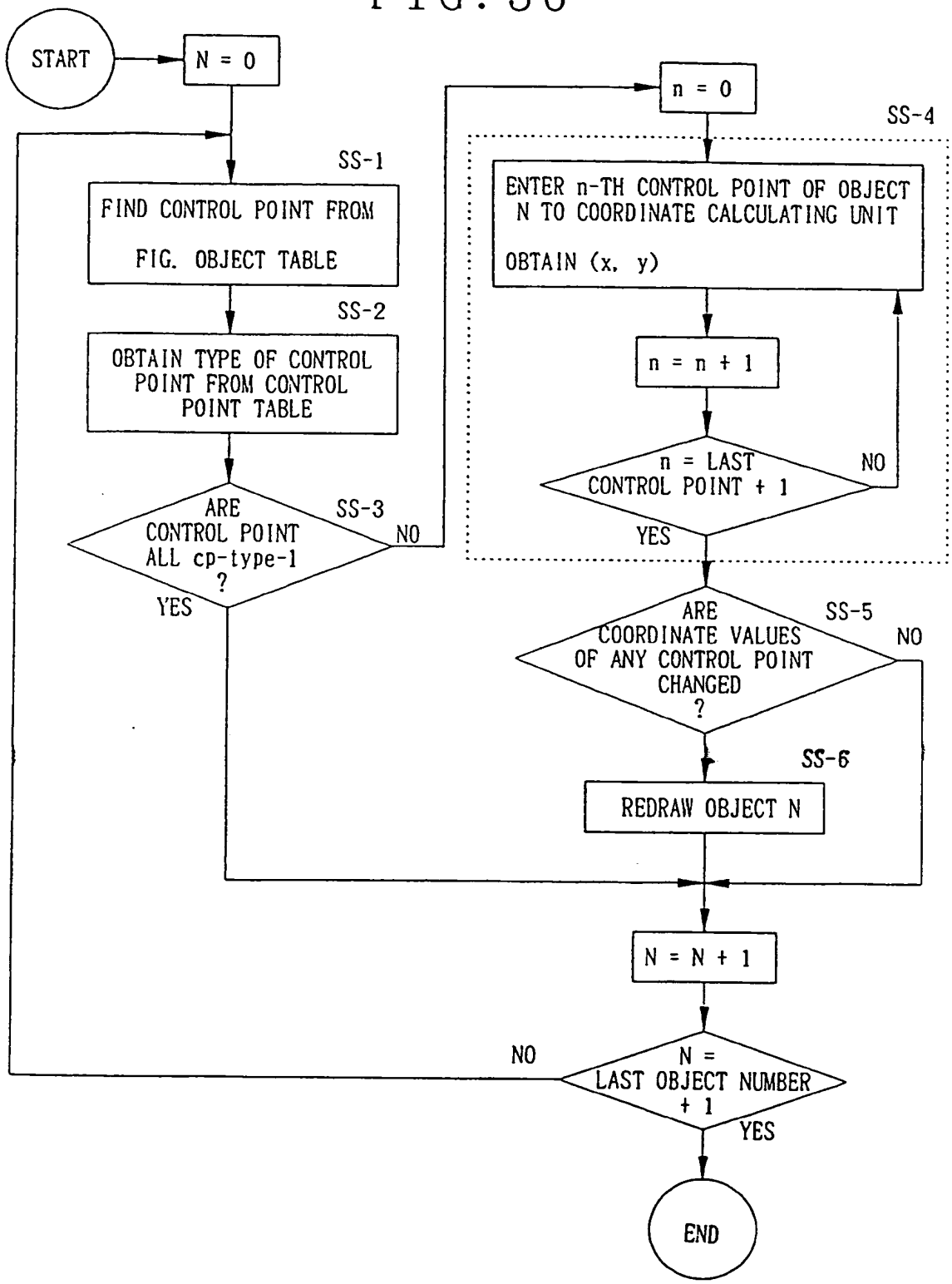
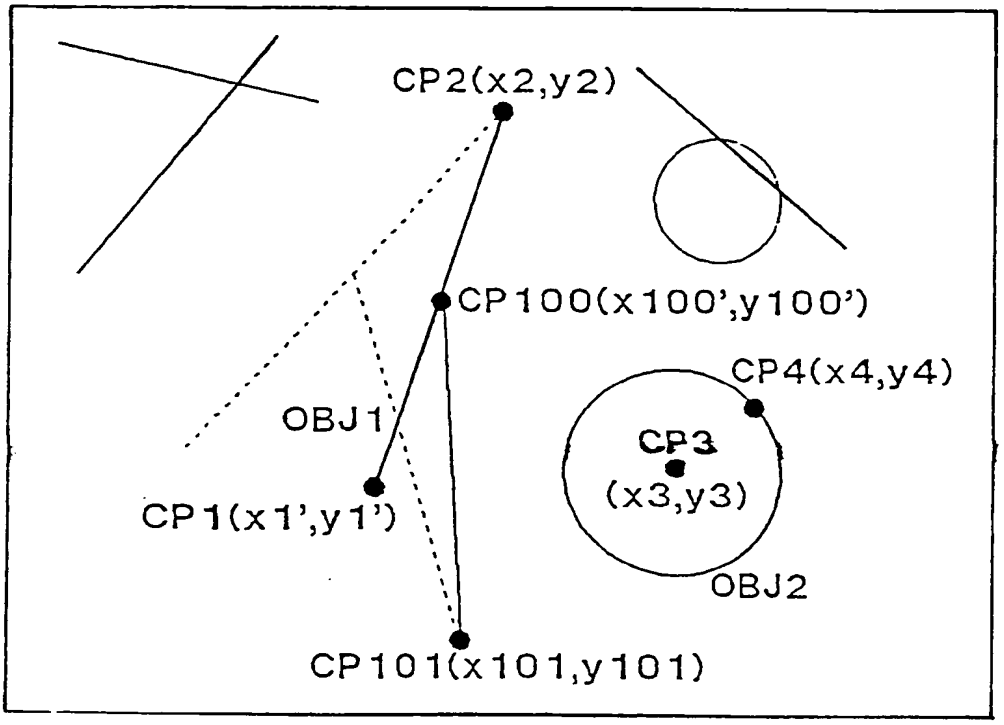

START
N = 0
SS-1
FIND CONTROL POINT FROM FIG. OBJECT TABLE
SS-2
OBTAIN TYPE OF CONTROL POINT FROM CONTROL POINT TABLE
SS-3
ARE CONTROL POINT ALL cp-type-1 ?
NO
YES
n = 0
SS-4
ENTER n-TH CONTROL POINT OF OBJECT N TO COORDINATE CALCULATING UNIT
OBTAIN (x, y)
n = n + 1
n = LAST CONTROL POINT + 1
NO
YES
SS-5
ARE COORDINATE VALUES OF ANY CONTROL POINT CHANGED ?
NO
SS-6
REDRAW OBJECT N
N = N + 1
N = LAST OBJECT NUMBER + 1
NO
YES
END

FIG. 37


CP2(x2,y2)
CP100(x100',y100')
CP4(x4,y4)
OBJ1
CP3
(x3,y3)
CP1(x1',y1')
OBJ2
CP101(x101,y101)

(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) Publication number: **0 492 901 A3**

(12) **EUROPEAN PATENT APPLICATION**

(21) Application number: **91311524.2**

(51) Int. Cl.5: **G06F 15/72**, G06F 3/033

(22) Date of filing: **11.12.91**

(30) Priority: **27.12.90 JP 415177/90**

(43) Date of publication of application:
01.07.92 Bulletin 92/27

(84) Designated Contracting States:
CH DE ES FR GB IT LI NL SE

(88) Date of deferred publication of the search report:
31.03.93 Bulletin 93/13

(71) Applicant: **International Business Machines Corporation**
Old Orchard Road
Armonk, N.Y. 10504(US)

(72) Inventor: **Hidaka, Kazuyoshi**
2-177 Fujidana-cho, Nishi-ku
Yokohama-shi, Kanagawa-ken(JP)

(74) Representative: **Blakemore, Frederick Norman**
IBM United Kingdom Limited Intellectual Property Department Hursley Park
Winchester Hampshire SO21 2JN (GB)

(54) Apparatus for and method of handling graphics objects.

(57) A graphics system comprises a graphics object table 10 that stores, for each graphics object, its identifier and the identifiers of one or more control points defining the graphics object; a control point table 11 that stores, for each control point, its identifier and positional data, said positional data being permitted to be data on the position of the control point relative to a graphics object; and means of updating the graphics object table 10 and the control point table 11 in response to the setting of one or more control points for a new graphics object by the user. These tables are referred to in response to a movement or reshaping of a primary graphics object, and the coordinate values of the affected control points are calculated by using a coordinate calculating unit 7.

Even when a number of graphics objects are interrelated, a graphics object under a geometical constraint exerted directly or indirectly by another graphics object is redrawn in response to the movement or reshaping of that object. Further, such redrawing can be done in response to a movement or reshaping other than translation, rotation, or scaling.

EP 0 492 901 A3

European Patent Office

EUROPEAN SEARCH REPORT

Application Number

EP 91 31 1524

DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (Int. Cl.5)
A	EUROGRAPHICS'88. PROCEEDINGS OF THE EUROPEAN COMPUTER GRAPHICS CONFERENCE AND EXHIBITION 1988, pages 233 - 242 VAN EMMERIK 'a system for graphical interaction on parameterized models' * paragraph 2 * * paragraph 3 * * paragraph 5 * * figure 3 * ---	1-17	G06F15/72 G06F3/033
A	EP-A-0 205 966 (IBM) * column 7, line 50 - column 8, line 49; figures 1-3 * ---	1-17	
A	EP-A-0 194 442 (IBM) * column 2, line 25 - column 2, line 61; figure 5 * ---	1-10	
A	US-A-4 821 214 (SEDERBERG) -----		
			TECHNICAL FIELDS SEARCHED (Int. Cl.5)
			G06F

The present search report has been drawn up for all claims

Place of search	Date of completion of the search	Examiner
THE HAGUE	29 JANUARY 1993	PEREZ MOLINA E.

CATEGORY OF CITED DOCUMENTS

X : particularly relevant if taken alone
Y : particularly relevant if combined with another document of the same category
A : technological background
O : non-written disclosure
P : intermediate document

T : theory or principle underlying the invention
E : earlier patent document, but published on, or after the filing date
D : document cited in the application
L : document cited for other reasons

& : member of the same patent family, corresponding document

EPO FORM 1503 03.82 (P0401)

This Page is Inserted by IFW Indexing and Scanning Operations and is not part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ BLACK BORDERS
- ☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☐ FADED TEXT OR DRAWING
- ☑ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☑ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: _____

IMAGES ARE BEST AVAILABLE COPY.
As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.